Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Прикладной Математики и Информатики
Научно-образовательный центр когнитивного моделирования

Направление подготовки / специальность: 09.04.01 Информатика и вычислительная техника
Направленность (профиль) подготовки: Методы и технологии искусственного интеллекта

СЕМАНТИЧЕСКАЯ РЕКОНСТРУКЦИЯ 3D-СЦЕН В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ ГАУССОВСКОГО СПЛАТТИНГА

(магистерская диссертация)

Студент:
Титков Роман Евгеньевич

_____
(подпись студента)

Научный руководитель:
Юдин Дмитрий Александрович,
канд. техн. наук

_____
(подпись научного руководителя)

Консультант *(при наличии)*:
_____
_____
(подпись консультанта)

Москва 2025

Содержание

1 Введение **4**

2 Обзор литературы и постановка задачи **6**

2.1 Методы представления и реконструкции трехмерных сцен 6

2.2 Методы семантической сегментации с открытым словарем 11

2.3 Современные методы SLAM 15

2.4 Выявленные пробелы и мотивация исследования 19

2.5 Постановка задачи и цель работы 20

3 Метод семантической реконструкции 3D-сцены **24**

3.1 Общая архитектура системы 25

3.2 Модуль извлечения DINOv2 признаков 28

3.3 Модуль сжатия семантических признаков 29

3.4 Интеграция семантических признаков в Gaussian Splatting 30

3.5 Модуль SLAM 31

3.6 Семантическая сегментация с открытым словарём 33

3.7 Рендеринг объектов по запросу с новых ракурсов 35

4 Экспериментальная часть **38**

4.1 Датасеты 38

4.2 Визуализация результатов экспериментов 41

4.3 Сравнение с существующими методами 44

4.4 Анализ влияния компонентов метода 47

5 Заключение **53**

Аннотация

Алгоритмы семантической 3D-реконструкции играют ключевую роль в системах компьютерного зрения автономных транспортных средств, обеспечивая глубокое понимание сцены. В работе представлен метод LEG-SLAM на основе Gaussian Splatting, который интегрирует семантические признаки DINOv2 и поддерживает сегментацию на основе запросов с открытым словарём. Для ускорения вычислений применяется PCA-сжатие эмбеддингов, а единая функция потерь учитывает цветовую, глубинную и семантическую составляющие. Эксперименты на публичных датасетах Replica и ScanNet демонстрируют высокую точность семантической сегментации и фотореалистичную визуализацию в режиме реального времени. Предложенный подход эффективно балансирует между качеством реконструкции и производительностью, что делает его перспективным для задач автономного вождения.

1 Введение

Задача реконструкции трёхмерной карты окружающего мира является одной из ключевых в мобильной робототехнике и автономном вождении. Такие карты необходимы для точной оценки позы робота, планирования траекторий и повышения надёжности навигации в динамичных и незнакомых условиях. Семантические карты — модели, которые помимо геометрии содержат информацию о классах объектов — позволяют повысить информативность и адаптивность робота: различать виды поверхностей, распознавать объекты инфраструктуры и учитывать их свойства при принятии решений.

Существует несколько способов представления трёхмерных сцен: облака точек, воксели, полигональные меши и неявные нейронные представления. Меш-карты удобны тем, что хранят топологию и могут включать различные атрибуты, такие как цвет, нормали, текстуры и семантика. Однако реконструкция плотных полигональных сетей требует значительных вычислительных ресурсов и с трудом укладывается в жёсткие рамки реального времени. Аналитические методы, основанные, например, на поверхностях уровня, часто работают быстро, но оказываются чувствительны к шуму и ограничены выбранной моделью.

Нейросетевые подходы, такие как NeRF и Gaussian Splatting, создают компактные и выразительные представления сложных 3D-сцен. Их ключевое преимущество — способность восстанавливать геометрию и визуальные характеристики поверхности с высокой детализацией и фотореализмом за счёт оптимизации параметров в структуре нейросетевого декодера. Однако такие методы изначально не включают семантическую информацию и предъявляют высокие требования к вычислительным ресурсам, что существенно ограничивает их применение в задачах онлайн-реконструкции и реального времени.

Современные семантические SLAM-алгоритмы, как правило, рассматривают семантику лишь как дополнительный атрибут, присваиваемый после построения геометрии: вершины меша или точки облака получают метки классов, однако эти метки не участвуют в самом процессе реконструкции. Лишь отдельные работы пытаются интегрировать семантическую информацию для улучшения одометрии или замыкания петель, но такие подходы не обеспечивают полноценной совместной оптимизации геометрии и семантики. В результате задача построения по-настоящему интегрированных семантических 3D-карт остаётся открытой и крайне актуальной для развития

SLAM-систем нового поколения.

В разделе 2 даются ключевые определения и обзор современных методов трёхмерной реконструкции сцены: аналитических, неявных нейросетевых представлений и Gaussian Splatting. Рассматриваются подходы к семантической сегментации и метрики качества семантических карт.

В разделе 3 подробно будет рассмотрен предложенный подход, описана архитектура, новизна и детали реализации.

В разделе 4 приводятся описание используемых датасетов, результаты оценки качества 3D-реконструкции, семантической сегментации и показателей SLAM. Приводится сравнение с современными методами.

В результате выполнения данной работы были получены следующие результаты:

- Предложен подход к семантической реконструкции на базе Gaussian Splatting, объединяющий геометрию и семантику с открытым словарём.
- Предложена функция потерь, учитывающая семантическую информацию в процессе оптимизации сцены.
- Реализован модуль извлечения визуальных признаков на базе DINOv2 и их компактного сжатия с помощью PCA без существенной потери точности.
- Осуществлена интеграция языковых эмбеддингов в процесс построения трёхмерной карты.
- Проведены эксперименты на датасетах Replica и ScanNet, оценены метрики 3D-реконструкции, семантики и SLAM. Показано превосходство LEG-SLAM над базовыми подходами.

Полученные результаты демонстрируют, что предложенный подход эффективно решает задачу построения семантически обогащённых трёхмерных карт в реальном времени, сохраняя высокую точность геометрии и обеспечивая возможность проводить семантическую сегментацию с открытым словарём. Разработанный алгоритм может служить надёжной основой для мобильных роботов и автономных систем и открывает перспективу для дальнейшего совершенствования семантических SLAM-алгоритмов.

2 Обзор литературы и постановка задачи

В данной главе представлен анализ современных методов представления трехмерных сцен, семантической сегментации и систем одновременной локализации и построения карты (SLAM). Рассмотрены классические и нейронные подходы к представлению трехмерных данных, их преимущества и ограничения. Особое внимание уделено методам на основе неявных функций и явных примитивов, а также их интеграции с семантической информацией. На основе проведенного анализа сформулированы ключевые проблемы и ограничения существующих подходов, что позволило определить направления для разработки нового метода.

2.1 Методы представления и реконструкции трехмерных сцен

Трехмерная реконструкция сцен является фундаментальной задачей компьютерного зрения, которая заключается в создании точного цифрового представления реального окружения. Эта задача имеет решающее значение для широкого спектра приложений. Качество реконструкции напрямую влияет на эффективность последующего анализа сцены, включая семантическую сегментацию и навигацию. В последние годы развитие нейронных методов представления сцен привело к значительному прогрессу в этой области, позволив создавать более детальные и точные реконструкции при меньших вычислительных затратах.

В данном разделе рассмотрены классические методы представления и реконструкции трехмерных сцен и современные нейронные подходы.

Классические методы

Классические подходы к представлению трехмерных сцен включают три основных типа: воксели, облака точек и меши.

Воксели представляют собой дискретизацию пространства на кубические элементы. Такой подход эффективен для задач, где важна работа с объемом, например, в медицинской визуализации или симуляциях. Однако, из-за экспоненциального роста количества вокселей с увеличением разрешения, требования к памяти существенно возрастают, а детализация ограничена размером вокселя.

Облака точек описывают поверхность объекта как множество несвязанных точек с координатами (x, y, z) и, при необходимости, дополнительными атрибутами

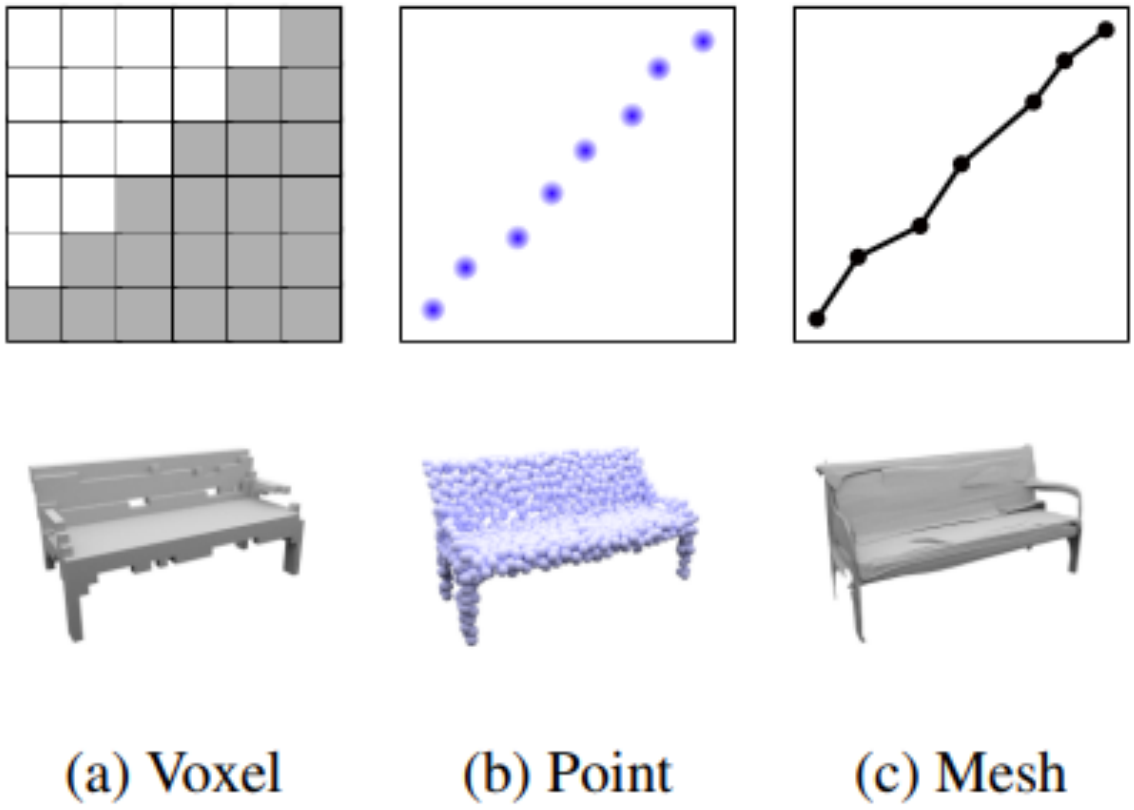


Рис. 1: Классические методы представления 3D-сцен

(цвет, нормали, семантика). Такой формат удобен для получения с помощью лидаров или стереокамер, но не содержит информации о топологии поверхности, что затрудняет задачи реконструкции и визуализации.

Меши строят поверхность объекта в виде связной сетки из полигонов. Меши позволяют точно моделировать сложные формы и поддерживают текстурирование, что делает их стандартом в компьютерной графике и системах автоматизированного проектирования. Однако построение и обработка мешей требуют сложных алгоритмов, а модификация топологии может быть трудоемкой.

Методы на основе нейронных сетей

Современные методы на основе нейронных сетей к представлению трехмерных сцен можно разделить на два основных направления: основанные на неявных функциях и на явных примитивах.

Ключевое различие между этими подходами показано на рисунке 2. NeRF (a) использует MLP для моделирования непрерывного поля плотности и цвета вдоль каждого луча, что позволяет получать высокую детализацию, но требует трасси-

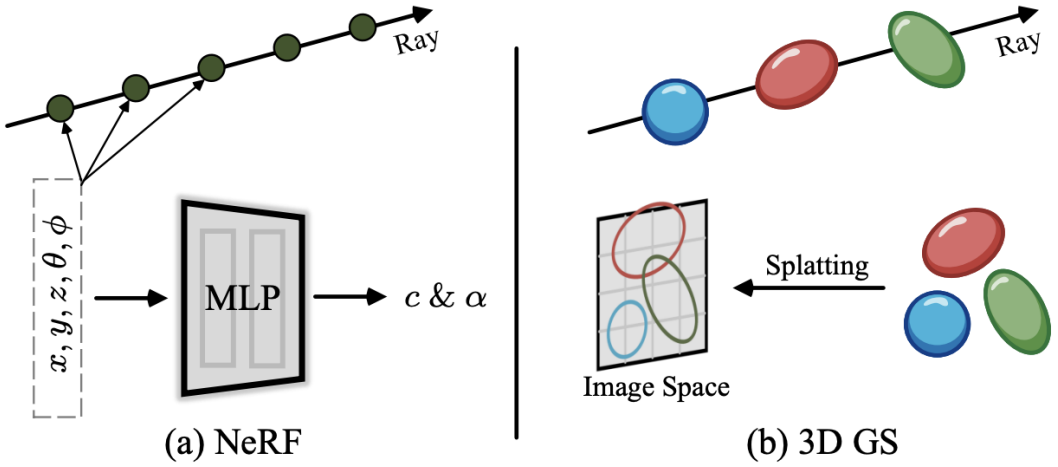


Рис. 2: Сравнение NeRF и Gaussian Splatting

ровки множества точек и сложных вычислений. В отличие от этого, 3D Gaussian Splatting (b) представляет сцену как множество параметрических гауссианов, которые напрямую проецируются в пространство изображения с помощью гауссового распределения, что обеспечивает высокую скорость визуализации и эффективную обработку сложных сцен.

NeRF

Neural Radiance Fields (NeRF) [1] — это метод, который моделирует трехмерную сцену как непрерывное поле плотности и цвета с помощью нейронной сети. Основная идея заключается в аппроксимации функции F:

$$F : (\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma), \tag{2.1}$$

где $\mathbf{x} = (x, y, z)$ — трехмерные координаты точки в пространстве, $\mathbf{d} = (\theta, \varphi)$ — направление луча, $\mathbf{c}$ — цвет в формате RGB, σ — плотность (непрозрачность).

Для получения изображения сцены с новой точки наблюдения используется уравнение объемного рендеринга:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\, \sigma(\mathbf{r}(t))\, \mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt, \tag{2.2}$$

где $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ — параметризация луча, $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$ — накопленное затухание вдоль луча.

На практике интеграл аппроксимируется суммированием по дискретным точкам

вдоль луча:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \tag{2.3}$$

где $T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$, δ_i — расстояние между соседними точками.

NeRF обучается по множеству изображений с известных позиций камеры, минимизируя разницу между синтезированными и реальными изображениями:

$$\mathcal{L} = \sum_{\text{лучи}} \|\hat{C}(\mathbf{r}) - C^*(\mathbf{r})\|^2, \tag{2.4}$$

Данный подход позволяет получать фотореалистичные изображения с новых ракурсов, но требует значительного количества выборок вдоль каждого луча и существенных вычислительных ресурсов. Модификации и улучшения NeRF направлены на ускорение визуализации, добавление семантической информации, поддержку динамических сцен и другие аспекты, сохраняя при этом базовые принципы метода.

Среди наиболее заметных расширений NeRF можно отметить Instant-NGP [2], который ускоряет обучение на два порядка благодаря хэш-кодированию координат и смешанным точным вычислениям на GPU. Дальнейшее повышение интерактивности обеспечил Nerfacto [3]: объединение плотной воксельной сетки с полем радианса позволило получать изображения практически в реальном времени без ощутимой потери качества. Проблему динамики сцены решает Nerfies [4], вводящий деформационную модель пространства и тем самым позволяющий реконструировать объекты, меняющие форму или положение во времени. При дефиците обучающих данных эффективной оказалась DietNeRF [5], использующая самообучение и регуляризацию для устойчивой работы по нескольким кадрам. Наконец, Semantic-NeRF [6] расширяет оригинальную модель семантическими векторами, что делает возможной совместную оптимизацию геометрии, цвета и семантики в едином поле радианса.

Дополнительно, в контексте open-vocabulary сегментации были предложены методы LERF [7] и PVLFF [8]. Оба подхода строят NeRF-подобную реконструкцию и обучают визуально–языковое пространство по CLIP/LSeg-эмбеддингам, что позволяет выполнять поиск и сегментацию объектов по текстовым запросам непосредственно в 3D-сцене.

3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [9] — это современный подход к представлению и визуализации трехмерных сцен, основанный на явных примитивах. В отличие от

NeRF, который моделирует сцену как непрерывное поле, 3DGS описывает её как множество параметрических трехмерных гауссианов:

$$\mathcal{G} = \{(\mu_i, \Sigma_i, \mathbf{c}_i, \alpha_i)\}_{i=1}^{N}, \tag{2.5}$$

где $\mu_i \in \mathbb{R}^3$ — центр гауссиана, $\Sigma_i \in \mathbb{R}^{3\times3}$ — ковариационная матрица (определяет форму и ориентацию), $\mathbf{c}_i$ — цвет, α_i — непрозрачность.

Процесс визуализации состоит из следующих этапов:

1. **Проецирование:** Каждый трехмерный гауссиан проецируется в пространство изображения с учетом параметров камеры, что дает эллипс в двумерном пространстве.

2. **Сортировка:** Гауссианы сортируются по глубине относительно камеры для корректного альфа-смешивания.

3. **Распределение:** Каждый эллипс распределяется по пикселям изображения с весами, определяемыми гауссиановой функцией:

$$w_{ij} = \exp\left(-\frac{1}{2}(\mathbf{p}_j - \mu_i')^T \Sigma_i'^{-1}(\mathbf{p}_j - \mu_i')\right), \tag{2.6}$$

   где $\mathbf{p}_j$ — координата пикселя, μ_i', Σ_i' — центр и ковариация гауссиана в проекции.

4. **Альфа-смешивание:** Цвет и непрозрачность каждого гауссиана аккумулируются по стандартной формуле:

$$\mathbf{C}_j = \sum_i \alpha_{ij}\mathbf{c}_i \prod_{k<i}(1 - \alpha_{kj}), \tag{2.7}$$

   где α_{ij} — вклад i-го гауссиана в j-й пиксель.

Параметры гауссианов оптимизируются градиентным спуском по фотометрической ошибке между синтезированными и реальными изображениями, аналогично NeRF:

$$\mathcal{L} = \sum_{\text{пиксели}} \|\mathbf{C}_j - \mathbf{C}_j^*\|^2, \tag{2.8}$$

Главное преимущество 3DGS — высокая скорость визуализации и возможность эффективно моделировать сложные сцены с фотореалистичной детализацией. Метод легко расширяется для поддержки динамических сцен, семантической информации и текстового управления (например, LangSplat [10], Semantic Gaussians [11]).

Дальнейшее развитие 3DGS охватывает несколько ключевых направлений. MotionGS [12] вводит временные параметры в описание гауссианов, позволяя воспроизводить плавные деформации динамических сцен. Для интерактивного редактирования сцен был предложен RT-GS2 [13], в котором оптимизированы этапы сортировки и рендеринга. Семантическую составляющую развивает работа Semantic Gaussians [11]: каждому примитиву сопоставляется компактный семантический вектор, и сцена обучается одновременно по фотометрической и семантической ошибкам, что позволяет выполнять текстовый поиск непосредственно в 3D-пространстве. Иной подход реализует LangSplat [10], где поверх реконструированной RGB-сцены создаётся отдельная семантическая сцена; оптимизации подвергаются лишь три семантических коэффициента, согласованные с пространством CLIP-эмбеддингов. Работа Feature3DGS [14] сочетает 3DGS с LSeg-эмбеддингами и показывает заметный прирост точности open-vocabulary сегментации при умеренных вычислительных затратах.

2.2 Методы семантической сегментации с открытым словарем

Семантическая сегментация является ключевой задачей компьютерного зрения, позволяющей присваивать каждому пикселю изображения семантическую метку. В последние годы особый интерес вызывает направление сегментации с открытым словарем (open-vocabulary segmentation), которое позволяет работать с произвольными классами объектов, не ограничиваясь предопределенным набором категорий.

Основные проблемы и ограничения

- **Масштабируемость:** Традиционные методы требуют переобучения для новых классов, что делает их непрактичными для реальных приложений.
- **Согласованность:** Обеспечение согласованности сегментации между различными ракурсами и временными срезами.
- **Вычислительная эффективность:** Баланс между точностью сегментации и скоростью обработки.
- **Устойчивость к шуму:** Управление неоднозначностью в сложных сценах и при неблагоприятных условиях освещения.

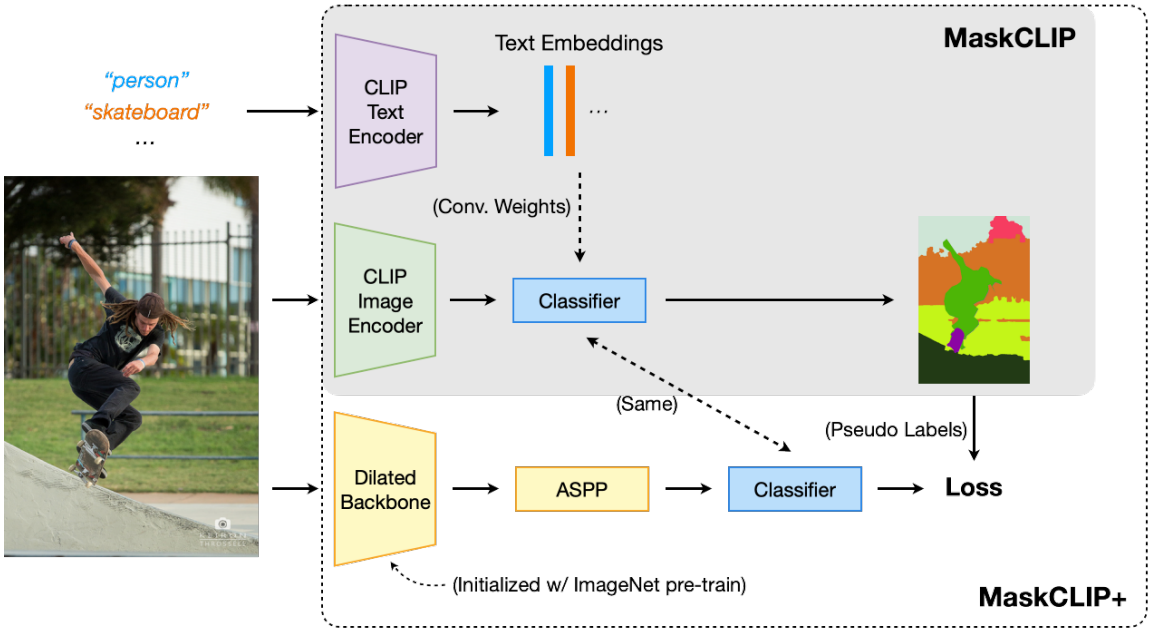


Рис. 3: Архитектура MaskCLIP+ [17]

Методы на основе отдельных изображений

Современные методы сегментации с открытым словарем, работающие с отдельными изображениями, можно условно разделить на два направления: подходы, основанные на CLIP-эмбеддингах, и методы, использующие DINO-эмбеддинги.

CLIP-эмбеддинги. Ранние работы, такие как OpenSeg [15] и LSeg [16], используют CLIP-подобные эмбеддинги для сегментации произвольных классов на отдельных изображениях. Принцип их работы заключается в сопоставлении визуальных признаков изображения с текстовыми описаниями классов, что позволяет выделять объекты, не встречавшиеся в обучающей выборке. Однако такие методы сталкиваются с рядом ограничений: они не обеспечивают временной и пространственной согласованности, что приводит к нестабильным результатам при обработке последовательностей изображений или различных ракурсов одной сцены; кроме того, высокая чувствительность к шуму и неоднозначностям в изображениях негативно сказывается на качестве сегментации, особенно в сложных условиях. Производительность подобных подходов также заметно снижается на сложных сценах с большим количеством объектов и разнообразием текстур.

В качестве примера можно привести работу MaskCLIP+, которая также использует CLIP-эмбеддинги для сегментации с открытым словарем. MaskCLIP+ [17] реализует двухэтапный подход: сначала выделяются потенциальные маски объектов на

изображении, а затем каждая маска сопоставляется с текстовыми описаниями классов с помощью CLIP. Такой подход позволяет сегментировать объекты, не встречавшиеся в обучающей выборке, и расширяет возможности сегментации с открытым словарем.

DINO-эмбеддинги. Альтернативный подход основан на использовании эмбеддингов, полученных с помощью моделей семейства DINO, в частности DINOv2. Ярким примером такого метода является Talk2DINO [18], архитектура которого представлена на рис. 4. Данный подход позволяет использовать локальные признаки DINOv2 для более точной и детализированной сегментации объектов на изображении. Важно отметить, что эмбеддинги DINOv2 изначально не выровнены с текстовым пространством, поэтому для сопоставления визуальных признаков с текстовыми описаниями требуется специальный адаптер, который переводит текстовые запросы из пространства CLIP в пространство DINOv2. Именно такую функцию выполняет архитектура Talk2DINO, что позволяет ей достигать высокой точности сегментации, устойчивости к шуму и уверенной работы на сложных сценах, где требуется детальное выделение объектов и их границ.

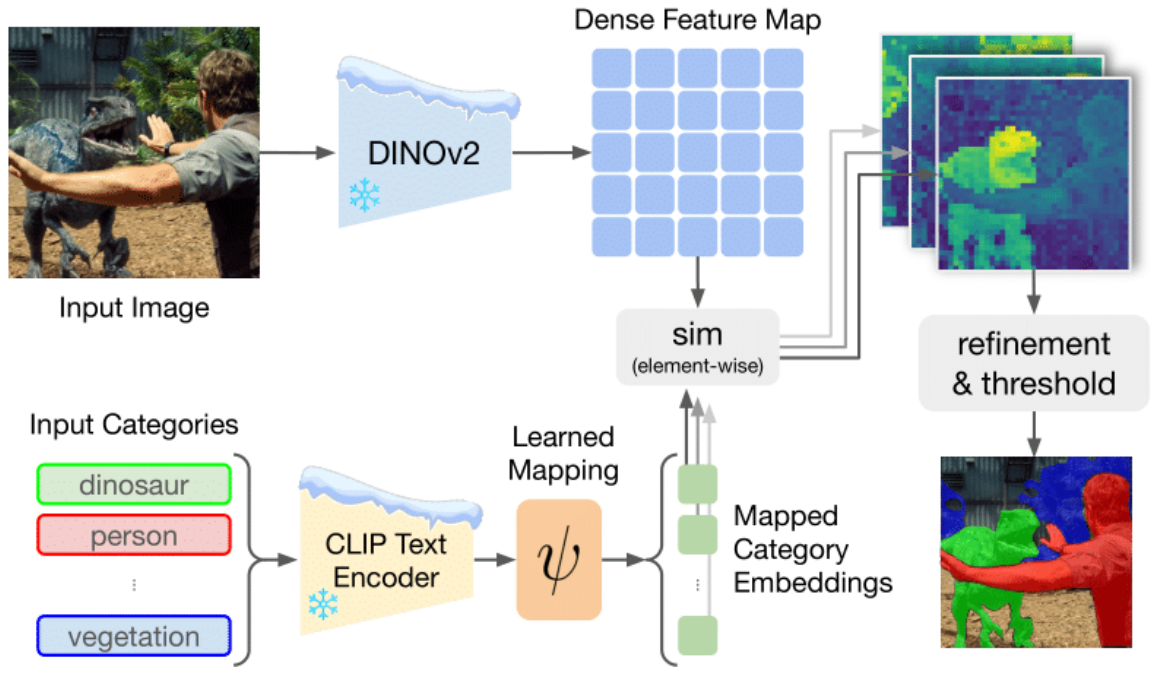


Рис. 4: Архитектура Talk2DINO [18]

Интеграция с 3D-представлениями

Трехмерная реконструкция сцен играет ключевую роль в обеспечении согласованности семантической сегментации между различными ракурсами и временными срезами. В отличие от методов, работающих с отдельными изображениями, 3D-реконструкция позволяет создавать единое семантическое представление сцены, которое сохраняет пространственную согласованность и устойчиво к изменениям точки наблюдения. Это особенно важно для задач навигации и взаимодействия с окружающей средой, где требуется стабильное понимание семантики сцены в реальном времени.

В последние годы наблюдается активное развитие методов, направленных на интеграцию семантической информации в 3D-представления сцен. Например, такие подходы, как Semantic Gaussians [11] и RT-GS2 [13], расширяют возможности 3D Gaussian Splatting за счёт добавления семантических признаков к каждому гауссиану. Это позволяет не только восстанавливать геометрию и текстуру сцены, но и обогащать её семантическими метками, что существенно расширяет спектр возможных приложений — от робототехники до дополненной реальности.

Отдельного внимания заслуживает подход LangSplat [10], который реализует альтернативную стратегию интеграции семантической информации. В отличие от методов, расширяющих гауссианы дополнительными признаками, LangSplat строит отдельную семантическую сцену, где пространственное расположение гауссианов фиксируется на основе RGB-реконструкции. При этом оптимизируется только семантический компонент, представленный тремя величинами, которые заменяют RGB-значения в процессе рендеринга. Такой подход позволяет эффективно работать с текстовыми запросами, сохраняя при этом геометрическую структуру исходной сцены.

Тем не менее, подобные методы сталкиваются с рядом серьёзных вызовов. Во-первых, высокая вычислительная сложность приводит к значительным затратам времени и ресурсов на обработку и рендеринг, особенно при работе с большими и детализированными сценами. Во-вторых, интеграция семантики требует увеличения объёма видеопамяти, что ограничивает масштабируемость подхода. Наконец, одновременная оптимизация геометрических и семантических параметров усложняет процесс обучения и может приводить к компромиссам между точностью реконструкции и качеством семантической сегментации.

2.3 Современные методы SLAM

Системы одновременной локализации и построения карты (SLAM) представляют собой комплексные решения, объединяющие задачи локализации робота в пространстве и создания трехмерной реконструкции окружающей среды. Качество 3D-реконструкции является критически важным аспектом современных SLAM-систем, так как оно определяет не только визуальное представление сцены, но и точность локализации, а также возможности последующего семантического анализа.

Интеграция нейронных представлений в системы одновременной локализации и построения карты (SLAM) является одним из наиболее перспективных направлений развития компьютерного зрения и робототехники. Традиционные SLAM-системы, основанные на разреженных ключевых точках или плотных картах глубины, сталкиваются с ограничениями в точности реконструкции, масштабируемости и способности к семантическому пониманию сцены. Нейронные представления открывают новые возможности для создания более точных, компактных и семантически богатых карт окружающей среды.

Эволюция представлений в нейронных SLAM-системах

Развитие нейронных SLAM-систем прошло значительный путь от первых неявных представлений до современных эффективных архитектур. Начальным этапом стали системы, основанные на неявных представлениях, где геометрия сцены кодировалась в весах нейронной сети. Пионерской работой в этой области стал Nice-SLAM [19], который интегрировал NeRF-подобные представления в SLAM-конвейер. Система использовала иерархическую сетку признаков для кодирования геометрии и цвета сцены, разделяя представление на глобальную и локальную компоненты. Глобальная сетка кодировала общую структуру сцены, в то время как локальные признаки обеспечивали детализацию в областях активного исследования. Однако такие системы сталкивались с ограничениями в скорости обработки и требовали значительных вычислительных ресурсов.

Следующим шагом в эволюции стали методы, использующие ускоренные структуры данных. Vox-Fusion [20] предложил адаптивную воксельную структуру для более эффективного кодирования пространственной информации, комбинируя преимущества воксельных и неявных представлений. Co-SLAM [21] внедрил многомасштабные хэш-сетки для кодирования сцены и реализовал глобальную оптимизацию без

необходимости выделения ключевых кадров. ESLAM [22] ввел концепцию энтропийной регуляризации для улучшения качества реконструкции и стабильности оптимизации, используя принципы теории информации для более эффективного распределения вычислительных ресурсов.

Настоящим прорывом стало появление методов на основе 3D Gaussian Splatting. SplaTAM [23] стал одной из первых успешных попыток интеграции данного представления в SLAM-пайплайн: авторы показали, что гауссиановое моделирование позволяет сохранять высокое качество реконструкции и при этом обеспечивать приемлемую скорость работы. В системе реализовано адаптивное управление примитивами: гауссианы динамически добавляются в области с новыми наблюдениями и удаляются при избыточности, а все параметры оптимизируются совместно методом градиентного спуска. GS-SLAM [24] развил эти идеи, предложив дополнительные механизмы регуляризации и глобального выравнивания, что повысило устойчивость трекинга и точность локализации. В RGBD GS–ICP SLAM [25] гауссиановое представление было объединено с классическим алгоритмом ICP, позволив достичь субметровой точности при умеренных вычислительных затратах.

Дальнейшие исследования значительно расширили функциональность подхода. Gaussian-SLAM [26] унифицировал задачи трекинга и плотной реконструкции в едином оптимизационном процессе по параметрам гауссианов, что улучшило метрики локализации без существенного снижения производительности. MonoGS [27] впервые продемонстрировал применимость метода к монокулярным RGB-последовательностям за счёт восстановления псевдоглубины. GS^{3}SLAM [28] ввёл трёхуровневую иерархию гауссианов, обеспечив более выгодный компромисс между качеством и скоростью. В OVO-Gaussian-SLAM [29] реализована поддержка открытого словаря семантических запросов при сохранении высоких показателей локализации и визуализации. Наконец, RGBDS-SLAM [30] предложил гибридную схему получения глубины, комбинируя данные RGB-D сенсоров и стерео-оценки, что повысило плотность и достоверность карты при закрытом словаре классов.

Современные нейронные SLAM-системы демонстрируют разнообразие архитектурных решений, направленных на оптимизацию баланса между качеством реконструкции и производительностью. Особое внимание уделяется гибридным подходам, которые комбинируют различные типы представлений для достижения оптимальных результатов. RTG-SLAM [31] представляет собой значимый шаг в направлении оптимизации производительности, используя продвинутые техники оптимизации и

параллелизации. Хотя система демонстрирует улучшенную производительность по сравнению с предыдущими методами, достижение полноценной работы в реальном времени остается сложной задачей.

Интеграция с классическими методами представляет собой еще одно ключевое направление развития. Современные системы активно используют проверенные временем алгоритмы, такие как ICP, в сочетании с нейронными представлениями. Это позволяет сохранить преимущества классических подходов в плане надежности и точности, одновременно получая выгоду от современных методов представления и рендеринга.

Семантические расширения

Интеграция семантической информации в нейронные SLAM-системы открывает новые возможности для понимания и интерпретации окружающей среды. SGS-SLAM [32] расширяет представление на основе гауссианов, добавляя семантические атрибуты к каждому примитиву. Система использует механизмы cross-attention для эффективного объединения визуальной и семантической информации, что позволяет достигать высокого качества сегментации при сохранении производительности.

Особое внимание уделяется работе с динамическими сценами. DNS-SLAM [33] фокусируется на обработке сцен с изменяющейся семантической структурой, обеспечивая адаптацию к появлению новых объектов и изменению семантических отношений в реальном времени. SemGauss-SLAM [34] использует низкоразмерные эмбеддинги для эффективного кодирования семантической информации, демонстрируя возможность достижения высокого качества сегментации при минимальных дополнительных вычислительных затратах.

Навигационные приложения представляют собой важное направление использования семантической информации. SNI-SLAM [35] интегрирует семантическую информацию для улучшения навигационных возможностей системы, позволяя роботам более эффективно планировать траектории и взаимодействовать с окружающей средой на основе семантического понимания сцены. NEDS-SLAM [36] вводит концепцию нейронных дескрипторов для более эффективного представления семантической информации, используя обучаемые дескрипторы, которые автоматически адаптируются к специфике конкретной сцены.

Сравнительный анализ и тенденции

Анализ современных нейронных SLAM-систем выявляет несколько ключевых тенденций в развитии области. Наблюдается явный переход от неявных представлений к явным, с методами на основе 3D Gaussian Splatting, демонстрирующими превосходство в скорости рендеринга. Гибридные архитектуры становятся стандартом, комбинируя различные представления для оптимизации баланса качество-производительность. Растет интерес к системам, способным к семантическому пониманию сцены, что открывает новые возможности для практических приложений.

Особое место среди современных методов занимает Photo-SLAM [37], который представляет собой комплексную систему, объединяющую точную локализацию с высококачественным фотореалистичным рендерингом. Система использует гибридное представление, сочетающее точечные облака для трекинга и гауссианы для рендеринга, что позволяет достигать оптимального баланса между качеством и производительностью. Photo-SLAM демонстрирует впечатляющие результаты в плане скорости работы и качества реконструкции, что делает его отличным базовым методом для дальнейших улучшений. Однако, как и многие современные SLAM-системы, он не включает в себя семантическую составляющую, что ограничивает его возможности в задачах, требующих понимания содержания сцены.

Несмотря на значительный прогресс, современные нейронные SLAM-системы по-прежнему сталкиваются с рядом вызовов. Обработка больших сцен и длинных траекторий остается вычислительно сложной задачей, требующей дальнейшей оптимизации. Стабильная работа в условиях плохого освещения, быстрых движений и динамических изменений сцены представляет собой серьезную проблему. Поддержание консистентности семантических представлений во времени и пространстве требует разработки новых механизмов. Эффективное объединение визуальной, геометрической и семантической информации остается актуальной задачей.

Перспективы развития области связаны с дальнейшей оптимизацией производительности, улучшением качества реконструкции и расширением семантических возможностей. Особое внимание уделяется разработке методов, способных работать в реальном времени на стандартном оборудовании, что критически важно для практических приложений. Развитие новых архитектур и алгоритмов оптимизации позволит преодолеть текущие ограничения и создать более эффективные системы для работы со сложными сценами.

2.4 Выявленные пробелы и мотивация исследования

Анализ современного состояния области выявляет несколько ключевых пробелов, которые ограничивают практическое применение существующих методов и определяют направления для дальнейших исследований.

Компромисс между качеством и производительностью. Современные SLAM-системы сталкиваются с фундаментальным компромиссом между качеством реконструкции и скоростью работы. Методы, обеспечивающие высокое качество визуализации и точную геометрическую реконструкцию, часто требуют значительных вычислительных ресурсов, что делает их неприменимыми для real-time приложений. В то же время, системы, оптимизированные для работы в реальном времени, часто жертвуют качеством реконструкции или стабильностью работы. Особенно остро эта проблема проявляется при работе со сложными сценами, содержащими множество деталей и динамических объектов. Photo-SLAM представляет собой редкое исключение, демонстрируя впечатляющие результаты как в плане качества реконструкции, так и в скорости работы, достигая производительности, необходимой для real-time приложений. Однако его основной недостаток заключается в отсутствии семантического понимания сцены, что ограничивает его применение в задачах, требующих интерпретации содержания.

Ограниченная интеграция семантической информации. Несмотря на значительный прогресс в области семантической сегментации с открытым словарем, большинство существующих SLAM-систем либо работают с фиксированным набором классов, либо интегрируют семантическую информацию с существенными компромиссами в производительности. Методы на основе CLIP-эмбеддингов, обеспечивающие гибкость в работе с произвольными классами, требуют значительных вычислительных ресурсов, что делает их непрактичными для real-time приложений. В то же время, более эффективные подходы, такие как Photo-SLAM, не включают в себя семантическую составляющую, что ограничивает их возможности в задачах, требующих понимания содержания сцены.

Неэффективное использование вычислительных ресурсов. Современные подходы часто дублируют вычисления, выполняя семантическую сегментацию и геометрическую реконструкцию в отдельных пайплайнах. Это приводит к неоптимальному использованию GPU-памяти и вычислительных ресурсов, особенно критичному для мобильных и встраиваемых систем. Кроме того, существующие мето-

ды не в полной мере используют преимущества современных архитектур нейронных сетей и техник оптимизации, что приводит к избыточным вычислениям и неэффективному использованию доступных ресурсов.

Ограничения в работе с динамическими сценами. Большинство современных SLAM-систем демонстрируют ограниченную способность к работе с динамическими сценами. Методы, основанные на предположении о статичности сцены, не могут корректно обрабатывать движущиеся объекты, что приводит к артефактам в реконструкции и ошибкам в оценке траектории. Существующие подходы к обработке динамических сцен либо требуют значительных вычислительных ресурсов, либо жертвуют качеством реконструкции, что делает их неприменимыми для практических задач.

Мотивация исследования. Выявленные пробелы определяют необходимость разработки нового подхода, который бы преодолел существующие ограничения и обеспечил эффективную интеграцию семантической информации в SLAM-системы. В качестве базового метода выбран Photo-SLAM, который демонстрирует отличные результаты в плане качества реконструкции и скорости работы, но не включает в себя семантическую составляющую. Основная цель исследования заключается в расширении возможностей Photo-SLAM за счет интеграции эффективных методов семантической сегментации с открытым словарем, что позволит создать систему, способную не только точно реконструировать геометрию сцены, но и понимать её семантическое содержание в реальном времени.

2.5 Постановка задачи и цель работы

Основываясь на выявленных пробелах в существующих методах, данная работа направлена на разработку эффективной системы семантической реконструкции 3D-сцен, которая преодолевает ограничения современных подходов. Ключевая цель заключается в создании метода, который одновременно обеспечивает высокое качество визуальной реконструкции сцены и позволяет проводить точную семантическую сегментацию по произвольным текстовым запросам в режиме реального времени.

Формальная постановка задачи

Входные данные:

- $\mathcal{D} = \{(I_t, D_t)\}_{t=1}^{N}$ — последовательность RGB-D кадров, где:

  - $I_t \in \mathbb{R}^{H \times W \times 3}$ — RGB-изображение в момент времени t
  - $D_t \in \mathbb{R}^{H \times W}$ — карта глубины

- $\mathcal{M} = \{M_t\}_{t=1}^{N}$ — GT маски семантических классов для каждого кадра из $\mathcal{D}$; $M_t \in \{0, \ldots, C\}^{H \times W}$, где C — количество рассматриваемых классов;
- $\mathcal{Q} = \{q_m\}_{m=1}^{N}$ — набор текстовых запросов для оценки качества семантической сегментации в обученной сцене.

Выходные данные:

- $\mathcal{G} = \{(\mu_i, \Sigma_i, c_i, \alpha_i, f_i)\}_{i=1}^{N_g}$ — 3D-реконструкция сцены в виде семантических гауссианов, где:
  - $\mu_i \in \mathbb{R}^3$ — центр i-го гауссиана
  - $\Sigma_i \in \mathbb{R}^{3 \times 3}$ — ковариационная матрица (форма и ориентация)
  - $c_i \in \mathbb{R}^3$ — RGB-цвет
  - $\alpha_i \in [0, 1]$ — непрозрачность
  - $f_i \in \mathbb{R}^K$ — семантический вектор признаков
- $\mathcal{T} = \{T_t\}_{t=1}^{N}$ — траектория камеры, где $T_t \in SE(3)$ — оценочная поза камеры в момент t
- $\hat{I}_t, \hat{D}_t, \hat{F}_t$ — реконструированные RGB-изображение, карта глубины и карта семантических признаков

Целевая функция потерь:

Оптимизация параметров гауссианов выполняется по трем основным компонентам:

$$\mathcal{L}_{total} = \mathcal{L}_{color} + \mathcal{L}_{depth} + \mathcal{L}_{semantic}, \tag{2.9}$$

где:

$$\mathcal{L}_{color} = \sum_{t=1}^{N} \left[(1 - \lambda_{ssim}) \ell_1(\hat{I}_t, I_t) + \lambda_{ssim}(1 - \text{SSIM}(\hat{I}_t, I_t)) \right], \tag{2.10}$$

$$\mathcal{L}_{depth} = \sum_{t=1}^{N} \ell_1(\hat{D}_t, D_t), \tag{2.11}$$

$$\mathcal{L}_{semantic} = \sum_{t=1}^{N} \mathcal{L}_{cos_sim}(\hat{F}_t, F_t), \tag{2.12}$$

- $\mathcal{L}_{color}$ — фотометрическая ошибка, сочетающая ℓ_1-норму и структурное сходство (SSIM) для высококачественной реконструкции цвета

- $\mathcal{L}_{depth}$ — ошибка реконструкции глубины через ℓ_1-норму для обеспечения геометрической точности

- $\mathcal{L}_{semantic}$ — семантическая ошибка, измеряемая через косинусное сходство между рендеренными семантическими признаками и эталонными DINOv2 признаками. Это обеспечивает способность системы воспроизводить семантические признаки, которые затем используются для сегментации по текстовым запросам

Метрики оценки качества:

Визуальные метрики:

- **PSNR:** $10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right)$, — измеряет степень искажения реконструированного изображения;

- **SSIM:** оценивает структурное сходство изображений по яркости, контрасту и локальной структуре.

- **LPIPS** — оценивает перцептивную дистанцию в пространстве глубоких признаков

Семантические метрики:

- **Accuracy:** $\frac{\text{Correct}}{\text{Total}}$ — доля правильно классифицированных точек;

- **IoU:** $\frac{TP}{TP+FP+FN}$, **mIoU** — среднее по классам, мера перекрытия предсказанных и истинных масок (вычисляется, если доступны GT-маски $\mathcal{M}$).

Технические требования

Система должна функционировать в реальном времени и предоставлять возможность проводить сегментацию по запросам с открытым словарем. В рамках данной работы под реальным временем понимается способность производить реконструкцию сцены со скоростью не ниже 10 кадров в секунду.

Семантическая сегментация с открытым словарём

После обучения сцены система способна сегментировать объекты по произвольным текстовым запросам. Процесс включает три шага.

1. **Извлечение признаков.** После завершения обучения сцены для каждого кадра I_t из обученного облака гауссиан проецируется карта семантических признаков F_t с использованием оценённой позы камеры $T_t \in \mathcal{T}$. Таким образом, F_t представляет собой проекцию семантических признаков сцены в изображение.

2. **Формирование маски.** Текстовый запрос $q_m \in \mathcal{Q}$ обрабатывается модулем Talk2DINO: выход CLIP-текстового энкодера проходит через обучаемый адаптер, формируя эмбеддинг $z_m \in \mathbb{R}^D$. Затем применятся PCA, который проецирует z_m в пространство $\bar{z}_m \in \mathbb{R}^K$. Далее для каждого пикселя вычисляется косинусное сходство между $\bar{z}_m$ и соответствующим вектором из F_t. Полученная карта логитов преобразуется операцией argmax в предсказанную маску $\tilde{M}_{t,m}$.

3. **Валидация.** $\tilde{M}_{t,m}$ сравнивается с M_t, и рассчитываются Accuracy и IoU. При этом M_t не участвует в процессе оптимизации и служит только для оценки качества обученной сцены.

Интерактивный поиск в облаке гауссиан

После завершения оптимизации сцены пользователь взаимодействует с системой через набор запросов $\mathcal{Q}$. Для каждого запроса q_m необходимо:

- определить подмножество гауссиан $\mathcal{S}_m \subseteq \mathcal{G}$, максимально отвечающих запросу q_m;
- выполнить пространственную кластеризацию $\mathcal{S}_m$ и получить кластеры $\{\mathcal{C}_{m,k}\}$, каждый из которых интерпретируется как отдельный объект;
- для каждого кластера вычислить центроид $\boldsymbol{c}_{m,k} = \frac{1}{|\mathcal{C}_{m,k}|} \sum_{g \in \mathcal{C}_{m,k}} \boldsymbol{\mu}_g$, и ограничивающий параллелепипед с размерами $(w_{m,k}, d_{m,k}, h_{m,k})$ по осям сцены;
- вернуть пользователю набор описаний $\{(\boldsymbol{c}_{m,k}, w_{m,k}, d_{m,k}, h_{m,k})\}$ как ответ на запрос.

3 Метод семантической реконструкции 3D-сцены

В данной главе представлен детальный анализ разработанного метода семантической реконструкции 3D-сцены, который расширяет возможности Photo-SLAM за счет интеграции семантической информации. Предложенный подход позволяет не только эффективно восстанавливать геометрию и текстуру сцены, но и проводить семантическую сегментацию по произвольным текстовым запросам в режиме реального времени. Особое внимание уделено архитектурным решениям, обеспечивающим баланс между качеством реконструкции и производительностью, а также механизмам интеграции семантической информации в пайплайн гауссовского сплаттинга.

Общая схема метода представлена на рисунке 5. Метод основан на последовательной обработке входных RGB-D данных с извлечением и сжатием семантических признаков, их интеграцией в процесс гауссовского сплаттинга и последующей семантической сегментацией. Каждый этап оптимизирован для работы в реальном времени и обеспечивает высокое качество как реконструкции, так и семантического понимания сцены.

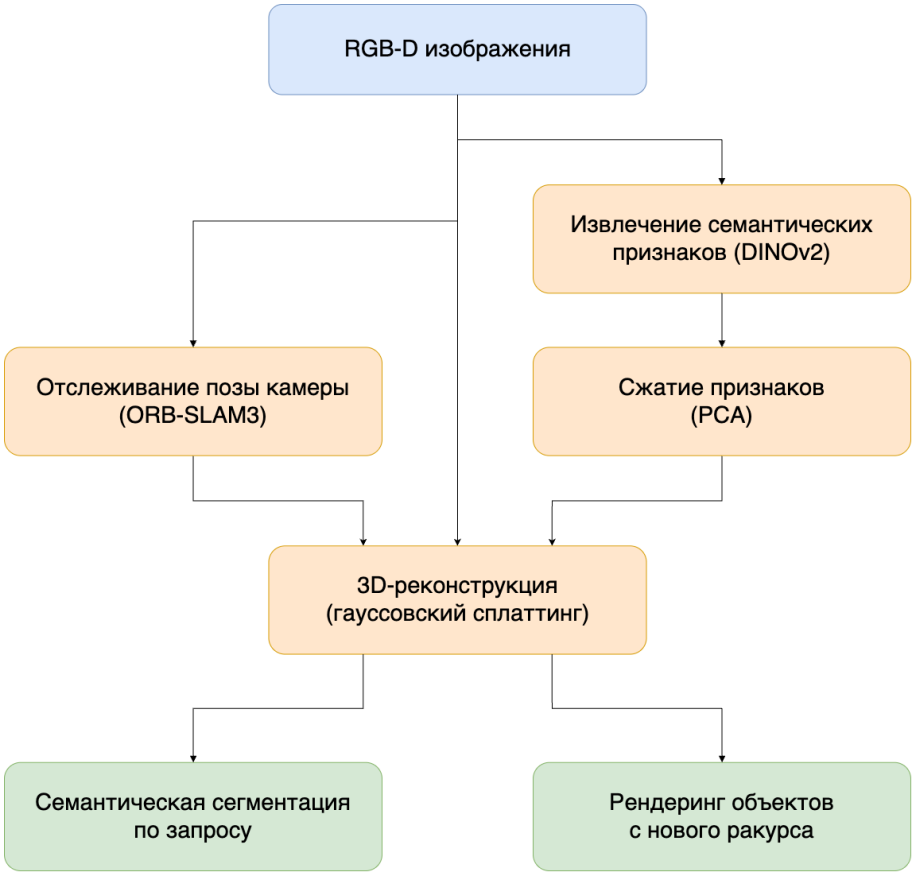


Рис. 5: Общая схема метода семантической реконструкции 3D-сцены

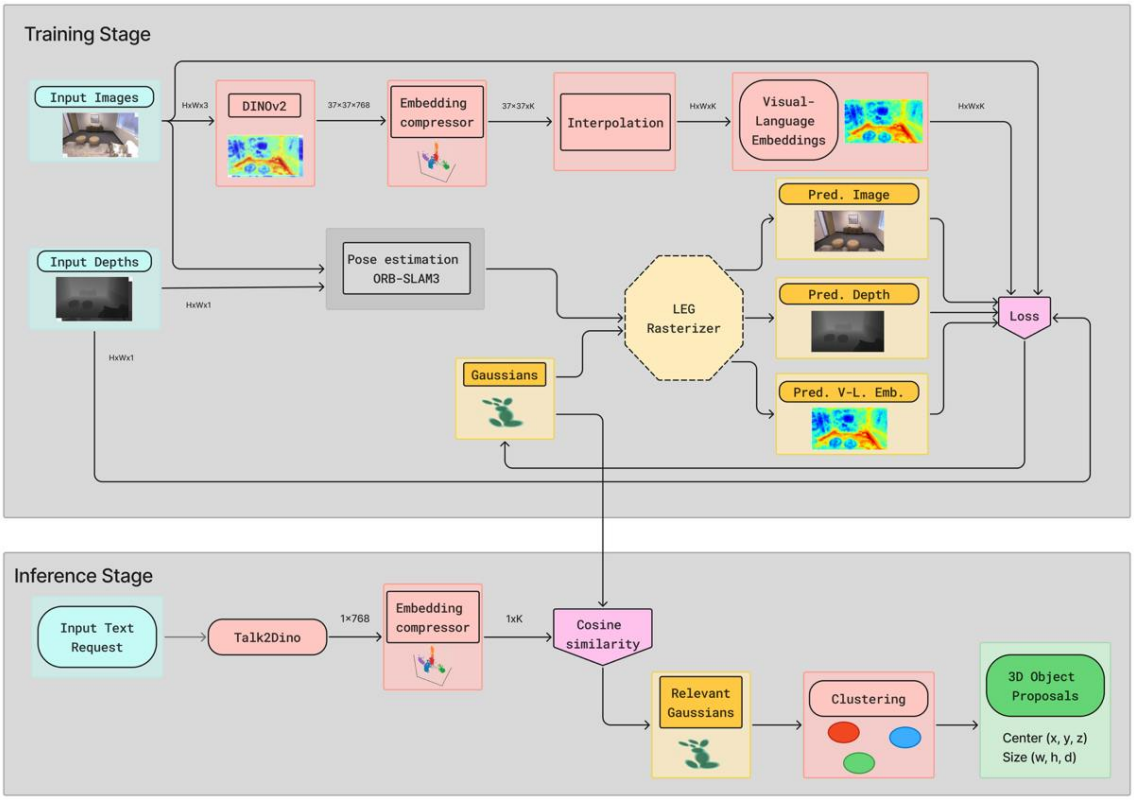


Рис. 6: Архитектура LEG-SLAM

Система реконструкции семантической 3D-сцены реализована на основе алгоритма Photo-SLAM [37] с модификацией для добавления семантической информации. Выбор Photo-SLAM в качестве базового метода обусловлен его высокой производительностью и качеством реконструкции, что делает его идеальной основой для расширения функциональности. Основные улучшения включают интеграцию DINOv2 для извлечения семантических признаков, эффективное сжатие эмбеддингов с помощью PCA и единый пайплайн оптимизации для геометрических и семантических параметров.

3.1 Общая архитектура системы

На рисунке 6 представлена архитектура модуля LEG–SLAM [38], включающая две ключевые фазы: этап обучения и этап инференса. В ходе обучения на вход системы поступает поток RGB–D кадров; выполняются оценка позы камеры, извлечение семантических признаков и совместная оптимизация параметров гауссиан, описывающих сцену. На этапе инференса оптимизированное гауссиановое представление

сопоставляется с текстовым запросом пользователя, что позволяет локализовать целевые объекты и вычислить координаты их центроидов.

Модуль отслеживания положения камеры основан на ORB-SLAM3 [39] и отвечает за оценку траектории камеры на основе входных RGB-D данных. Этот модуль обеспечивает стабильное отслеживание даже при быстрых движениях камеры и изменениях освещения, что критически важно для качественной реконструкции сцены.

Модуль извлечения семантических признаков использует предобученную модель DINOv2 [40], которая принимает входное изображение размером 518×518 пикселей и генерирует карту эмбеддингов размером $37 \times 37 \times 768$. В этой карте каждая из 37×37 ячеек соответствует локальному фрагменту сцены, а 768-мерный вектор представляет собой высокоуровневое описание его визуальных характеристик.

Модуль сжатия семантических признаков играет ключевую роль в оптимизации производительности системы, обеспечивая эффективное уменьшение размерности семантических признаков при сохранении их информативности. В качестве метода сжатия используется анализ главных компонент (PCA), который позволяет значительно снизить вычислительные затраты на последующих этапах обработки.

Модуль гауссовского сплаттинга оптимизирует 3D-сцену, представленную набором гауссиан. Каждая гауссиана кодирует не только геометрические и визуальные параметры, но и семантические признаки, что позволяет одновременно реконструировать сцену и проводить семантическую сегментацию.

Модуль семантической сегментации обеспечивает возможность выделения объектов по произвольным текстовым запросам. Он использует сжатые семантические признаки и механизмы выравнивания пространства признаков для эффективной работы с открытым словарем.

Взаимодействие между компонентами происходит следующим образом. Сначала входные RGB-D данные обрабатываются модулем отслеживания для оценки позы камеры. Одновременно из RGB-изображения извлекаются семантические признаки с помощью DINOv2, которые затем сжимаются через PCA для оптимизации производительности. Модуль гауссовского сплаттинга использует информацию о позе камеры и сжатые признаки для обновления и оптимизации 3D-представления сцены. Наконец, модуль семантической сегментации использует оптимизированное представление для выполнения запросов пользователя.

Такая архитектура обеспечивает эффективное использование вычислительных

ресурсов и высокое качество как реконструкции, так и семантической сегментации. Подробное описание каждого модуля и их взаимодействия представлено в следующих разделах.

Алгоритм 3.1 LEG-SLAM: Семантическая реконструкция 3D-сцены

Вход: Последовательность RGB-D изображений $\{I_t, D_t\}_{t=1}^{T}$

Вход: Настройки ORB-SLAM3, DINOv2, PCA компрессора

Выход: Семантическая 3D-карта гауссиан $\mathcal{G}$, траектория камеры $\mathcal{T}$

Инициализировать ORB-SLAM3, DINOv2, PCA компрессор
Инициализировать пустую карту гауссиан $\mathcal{G} = \emptyset$
$\mathcal{T} = \emptyset$ — Траектория камеры
для $t = 1$ **to** T
  $I_{gt} \leftarrow$ Загрузить RGB изображение I_t
  $D_{gt} \leftarrow$ Загрузить карту глубины D_t
  $F_{dino} \leftarrow \text{DINOv2}(I_{gt})$ — Извлечение семантических признаков: $37 \times 37 \times 768$
  $F_{compressed} \leftarrow \text{PCA}(F_{dino})$ — Сжатие признаков: $37 \times 37 \times K$
  $L_{gt} \leftarrow$ Билинейная интерполяция$(F_{compressed})$ — Увеличение до $H \times W \times K$
  $T_t \leftarrow \text{ORB-SLAM3.Track}(I_{gt}, D_{gt}, L_{gt})$ — Отслеживание позы камеры
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_t\}$
  если ORB-SLAM3.isKeyFrame(t) **то**
    $\mathcal{G} \leftarrow \text{UpdateGaussianMap}(\mathcal{G}, I_{gt}, D_{gt}, L_{gt}, T_t)$ — Обновление карты гауссиан
    для $iter = 1$ **to** N_{opt}
      $I_{pred}, L_{pred}, D_{pred} \leftarrow \text{Render}(\mathcal{G}, T_t)$ — Оптимизация параметров гауссиан
      $\mathcal{L} \leftarrow \text{ComputeLoss}(I_{pred}, L_{pred}, D_{pred}, I_{gt}, L_{gt}, D_{gt})$
      Обновить параметры гауссиан в $\mathcal{G}$ используя $\nabla\mathcal{L}$
return $\mathcal{G}, \mathcal{T}$

Алгоритм 3.1 представляет основной пайплайн предложенного метода. На каждом кадре система извлекает семантические признаки с помощью DINOv2, сжимает их через PCA и использует для отслеживания позы камеры в ORB-SLAM3. Для ключевых кадров выполняется обновление и оптимизация карты гауссиан с одновременным рендерингом RGB-изображений и семантических признаков.

3.2 Модуль извлечения DINOv2 признаков

Модуль извлечения DINOv2 признаков является ключевым компонентом системы, обеспечивающим генерацию семантических представлений входных изображений. В качестве базовой модели используется предобученная версия DINOv2 [40], которая демонстрирует выдающиеся результаты в задачах визуального представления благодаря самообучению на большом наборе данных.

Архитектура и параметры модели. В системе используется модель DINOv2 ViT-L/14, которая принимает входные изображения размером 518×518 пикселей. Модель основана на архитектуре Vision Transformer (ViT) и состоит из 24 слоев трансформера с размерностью скрытого состояния 1024. После обработки изображения модель генерирует карту эмбеддингов размером $37 \times 37 \times 768$, где каждая ячейка соответствует локальному фрагменту сцены размером 14×14 пикселей в исходном изображении.

Процесс извлечения признаков. Процесс обработки изображения в модуле состоит из следующих этапов:

1. Предобработка входного изображения: изменение размера до 518×518 пикселей с сохранением пропорций
2. Разбиение изображения на патчи размером 14×14 пикселей
3. Преобразование патчей в последовательность токенов через линейную проекцию
4. Обработка последовательности токенов через слои трансформера
5. Генерация карты эмбеддингов размером $37 \times 37 \times 768$

Интеграция в общий пайплайн. Карта эмбеддингов, полученная от DINOv2, передается в модуль компрессии признаков без изменения размеров. Это позволяет работать с компактным представлением сцены, снижая вычислительные затраты на последующих этапах. Для уменьшения размерности эмбеддингов используется PCA-компрессор, который сжимает их с 768 до K измерений, сохраняя при этом большую часть важной семантической информации.

Использование DINOv2 в качестве основы для извлечения семантических признаков обеспечивает высокое качество сегментации при сохранении возможности обработки в реальном времени. Предобученная модель не требует дополнительного

обучения на целевых данных, что упрощает развертывание системы и обеспечивает стабильную работу в различных условиях.

3.3 Модуль сжатия семантических признаков

Модуль сжатия семантических признаков играет ключевую роль в оптимизации производительности системы, обеспечивая эффективное уменьшение размерности семантических признаков при сохранении их информативности. В качестве метода сжатия используется анализ главных компонент (PCA), который позволяет значительно снизить вычислительные затраты на последующих этапах обработки.

Архитектура компрессора. Модуль реализует линейное преобразование, которое проецирует 768-мерные эмбеддинги DINOv2 в пространство меньшей размерности K. Преобразование задается матрицей весов $W \in \mathbb{R}^{K\times 768}$, которая оптимизирована для сохранения максимальной дисперсии данных. Значение K выбирается эмпирически на основе компромисса между степенью сжатия и качеством сохранения семантической информации.

Процесс обучения PCA. Модель PCA была обучена на текстовых эмбеддингах 1000 классов из ImageNet [41]. Для каждого класса был получен текстовый эмбеддинг через CLIP, спроецированный в латентное пространство DINOv2 с помощью Talk2DINO [18]. Для повышения робастности модели каждый эмбеддинг был дополнен шумом при сохранении косинусного сходства не менее 0.95 с оригиналом. Такая стратегия позволяет PCA эффективно сжимать эмбеддинги за пределами классов ImageNet, обучаясь на общем проектировании признаков, а не на конкретных представлениях классов.

Процесс сжатия. Процесс сжатия эмбеддингов включает три последовательных этапа. Сначала выполняется нормализация входных эмбеддингов через центрирование и масштабирование. Затем происходит линейное преобразование через матрицу весов PCA, что позволяет получить сжатые эмбеддинги заданной размерности K.

Интеграция в пайплайн. После сжатия карта эмбеддингов увеличивается до исходного разрешения изображения с помощью билинейной интерполяции. Это обеспечивает пространственную согласованность с входным изображением и позволяет эффективно использовать сжатые признаки в последующих этапах обработки. Полученное представление размером $H \times W \times K$ используется в пайплайне Gaussian

Splatting для совместного рендеринга RGB-изображений и семантических масок.

3.4 Интеграция семантических признаков в Gaussian Splatting

Интеграция семантических признаков в пайплайн Gaussian Splatting представляет собой ключевой аспект предложенного метода, позволяющий одновременно реконструировать геометрию сцены и проводить семантическую сегментацию. Этот подход обеспечивает эффективное объединение визуальной и семантической информации в едином представлении, что позволяет достичь высокой точности как в реконструкции, так и в сегментации.

Расширение параметров гауссиан. Каждая гауссиана в системе расширена дополнительным семантическим вектором, который кодирует информацию о содержимом соответствующего фрагмента сцены. Таким образом, полный набор параметров гауссианы включает геометрические параметры (позиция $\mu \in \mathbb{R}^3$, масштаб $s \in \mathbb{R}^3$, ориентация $R \in \mathbb{R}^{3\times3}$), визуальные параметры (цвет $c \in \mathbb{R}^3$, прозрачность $\alpha \in \mathbb{R}$) и семантические параметры (вектор признаков $f \in \mathbb{R}^K$).

Процесс оптимизации. Оптимизация параметров гауссиан происходит одновременно для всех компонентов, что обеспечивает согласованность между геометрией, визуальными характеристиками и семантической информацией. Процесс включает следующие этапы:

1. **Инициализация семантических векторов:** При создании новой гауссианы её семантическому вектору присваиваются нулевые значения. Если гауссиана получена путём расщепления существующей, то семантические признаки копируются из родительской гауссианы.

2. **Совместная оптимизация:** Параметры оптимизируются путём минимизации функции потерь, которая учитывает как визуальное соответствие, так и семантическую согласованность. Полная функция потерь включает три компонента:

   - $\mathcal{L}_{color}$ — стандартная функция потерь для цветовых признаков из исходного метода 3DGS, включающая L1-норму и SSIM:

   $$\mathcal{L}_{color} = (1 - \lambda)\mathcal{L}_1(I_{pred}, I_{gt}) + \lambda\mathcal{L}_{ssim}(I_{pred}, I_{gt}), \tag{3.1}$$

   - $\mathcal{L}_{cos_sim}$ — функция потерь для семантических признаков, основанная на косинусном сходстве между срендеренной картой семантических признаков и признаками из DINOv2

- $\mathcal{L}_1$ — функция потерь для оптимизации значения глубины

Таким образом, полная функция потерь имеет вид:

$$\mathcal{L}(I_{pred},\, L_{pred},\, D_{pred};\, I_{gt},\, L_{gt},\, D_{gt}) = \\ = \underbrace{(1-\lambda)\mathcal{L}_1(I_{pred},\, I_{gt}) + \lambda\mathcal{L}_{ssim}(I_{pred},\, I_{gt})}_{\mathcal{L}_{color}} + \\ + \mathcal{L}_{cos_sim}(L_{pred},\, L_{gt}) + \mathcal{L}_1(D_{pred},\, D_{gt}), \quad (3.2)$$

где I_{pred}, L_{pred}, D_{pred} — срендеренные изображение, карта языковых признаков и глубина соответственно, а I_{gt}, L_{gt}, D_{gt} — соответствующие эталонные значения.

3. **Многоуровневая оптимизация:** Для ускорения и улучшения качества реконструкции оптимизация производится на последовательно увеличивающихся разрешениях входных данных ($H_g \times W_g$). Уменьшение разрешения входных изображений выполняется с помощью гауссовой пирамиды. Такой подход позволяет быстро установить достаточно точные параметры у гауссиан, которые затем эффективно дообучаются на изображениях высокой размерности.

4. **Адаптивное управление:** Система динамически управляет гауссианами, добавляя новые примитивы в областях с высокой ошибкой реконструкции и удаляя избыточные гауссианы. При этом семантические признаки новых гауссиан инициализируются на основе ближайших существующих примитивов.

Рендеринг семантической информации. Процесс рендеринга семантической информации происходит параллельно с рендерингом RGB-изображения. Для каждого пикселя вычисляется взвешенная сумма вкладов всех видимых гауссиан, где веса определяются на основе расстояния до центра гауссианы и её параметров. Семантические признаки рендерятся аналогично цветовым компонентам, в результате чего получается карта семантических признаков размером $H \times W \times K$.

3.5 Модуль SLAM

В предложенном методе для решения задачи SLAM используется ORB-SLAM3 [39], современная система одновременной локализации и построения карты, которая обеспечивает высокоточное определение позы камеры в реальном времени. В

отличие от классических подходов, ORB-SLAM3 используется для трекинга — оценки текущей позы камеры на основе входных изображений, а его карта ключевых точек служит для инициализации гауссиан в методе Gaussian Splatting. Построение и обновление плотной 3D-карты сцены полностью реализуется с помощью метода Gaussian Splatting, независимо от дальнейшего обновления карты ключевых точек ORB-SLAM3.

Архитектура и компоненты. В системе используется трекинг-компонент ORB-SLAM3, который включает три основных модуля. Трекер отвечает за отслеживание положения камеры в реальном времени, используя ORB-признаки и оптимизацию по градиентному спуску. Локальный мэппер управляет локальной картой ключевых точек для поддержания трекинга и инициализации гауссиан. Глобальный мэппер обеспечивает переиспользование ранее посещенных мест для улучшения трекинга.

Взаимодействие с Gaussian Splatting. Интеграция ORB-SLAM3 с методом Gaussian Splatting реализована следующим образом. Сначала ORB-SLAM3 вычисляет траекторию камеры на основе входных RGB-D данных. Карта ключевых точек ORB-SLAM3 используется для инициализации новых гауссиан в соответствующих 3D-точках. Полученная траектория применяется для определения позы камеры при рендеринге и оптимизации гауссиан. Построение плотной карты сцены осуществляется в модуле Gaussian Splatting, независимо от дальнейшего обновления карты ключевых точек. Наконец, рендеринг визуальных и семантических данных выполняется на основе оптимизированных гауссиан.

Преимущества подхода. Разделение задач трекинга и реконструкции обеспечивает несколько ключевых преимуществ. Во-первых, достигается быстрый и точный трекинг благодаря проверенной архитектуре ORB-SLAM3. Во-вторых, обеспечивается эффективная инициализация гауссиан на основе карты ключевых точек. В-третьих, реализуется независимая плотная реконструкция сцены с помощью Gaussian Splatting. В-четвертых, достигается эффективное использование вычислительных ресурсов за счет разделения задач. Наконец, обеспечивается возможность семантической сегментации благодаря независимому модулю извлечения признаков.

Таким образом, предложенная система использует ORB-SLAM3 для трекинга позы камеры и инициализации гауссиан, в то время как все задачи реконструкции и семантической сегментации решаются независимо с помощью Gaussian Splatting. Такое разделение позволяет достигать высокой точности и производительности в

задаче семантической реконструкции сцен в реальном времени.

3.6 Семантическая сегментация с открытым словарём

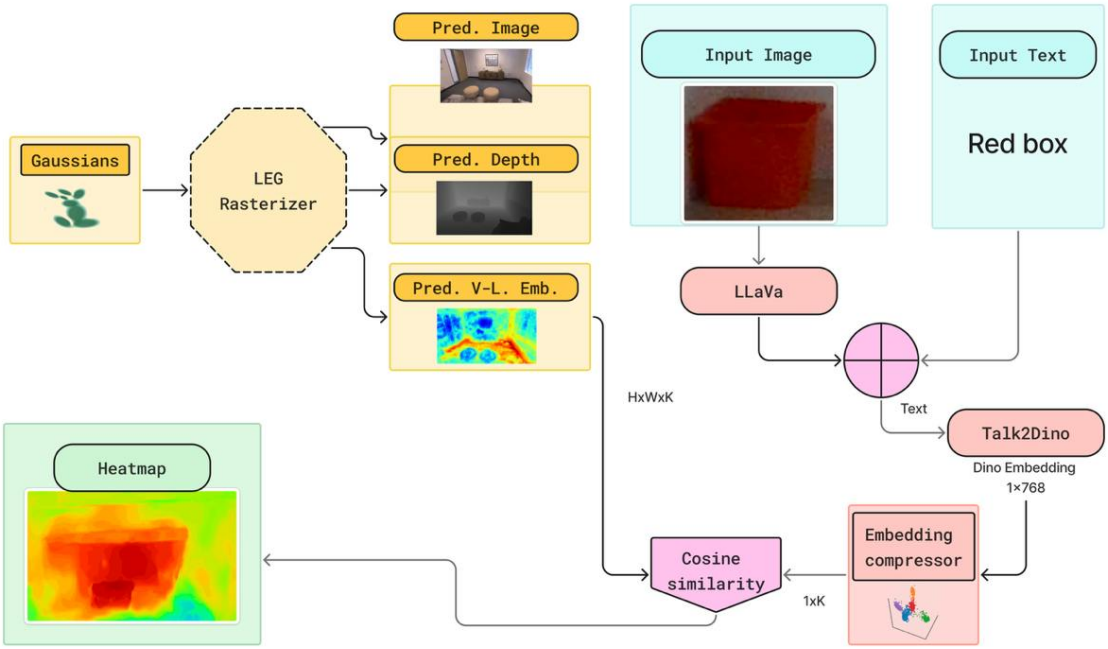


Рис. 7: Схема работы модуля семантической сегментации

Модуль семантической сегментации с открытым словарём позволяет выполнять сегментацию объектов по произвольным текстовым запросам без необходимости предварительного обучения на конкретных классах. Это достигается за счет использования предобученных моделей CLIP и Talk2DINO для преобразования текстовых запросов в пространство признаков DINOv2, которое используется в пайплайне Gaussian Splatting. Детальный процесс сегментации описан в алгоритме 3.2.

Архитектура и компоненты. Модуль включает следующие ключевые компоненты:

- **CLIP:** Преобразует текстовые запросы в эмбеддинги размерности 768
- **Talk2DINO:** Проецирует эмбеддинги CLIP в пространство признаков DINOv2
- **PCA:** Сжимает текстовые эмбеддинги до размерности K для соответствия визуальным признакам

- **LLaVa:** Опционально используется для генерации текстовых описаний из изображений

Интеграция с Gaussian Splatting. Сегментация тесно интегрирована с пайплайном Gaussian Splatting. Каждая гауссиана в сцене содержит сжатый семантический вектор размерности K, который хранит информацию о семантических признаках данного примитива. При рендеринге сцены семантические признаки гауссиан интерполируются по тем же правилам, что и цветовые компоненты, что позволяет получить на выходе карту семантических признаков размером $H \times W \times K$ для каждого пикселя изображения. Для поиска объектов, соответствующих текстовому запросу, полученная семантическая карта сравнивается с эмбеддингом запроса с помощью матричного умножения, как это подробно описано в алгоритме 3.2.

Алгоритм 3.2 Семантическая сегментация с открытым словарём

Вход: Текстовый запрос Q_{text} или изображение объекта I_{image}
Вход: Поза камеры T, реконструированная 3D-сцена $\mathcal{G}$
Вход: Модели CLIP, Talk2DINO, PCA, LLaVa (опционально)
Выход: Карта семантической сегментации $S \in \mathbb{R}^{H \times W}$

если входные данные — изображение I_{image} **то**
  $Q \leftarrow \text{LLaVa}(I_{image})$ — Генерация текстового описания
иначе
  $Q \leftarrow Q_{text}$ — Использование прямого текстового ввода
$E_{clip} \leftarrow \text{CLIP}(Q)$ — Обработка текстового запроса: 1×768
$E_{dino} \leftarrow \text{Talk2DINO}(E_{clip})$ — Выравнивание пространства признаков: 1×768
$E_{compressed} \leftarrow \text{PCA}(E_{dino})$ — Сжатие эмбеддинга: $1 \times K$
$I_{pred}, L_{pred}, D_{pred} \leftarrow \text{Render}(\mathcal{G}, T)$ — Проекция сцены на камеру
$S \leftarrow L_{pred} \cdot E_{compressed}$ — Матричное вычисление семантического сходства
$S \leftarrow \text{Нормализация}(S)$ — Приведение к диапазону [0,1]
return S

Предлагаемый метод семантической сегментации обладает рядом существенных научных преимуществ. Прежде всего, он обеспечивает возможность выделения объектов по произвольным текстовым запросам без необходимости дополнительного обучения на фиксированных классах, что принципиально расширяет спектр решаемых задач и позволяет работать с открытым семантическим словарём. Благодаря исполь-

зованию процедуры сжатия признаков достигается высокая вычислительная эффективность, что делает возможной обработку сцен в реальном времени и снижает требования к аппаратным ресурсам. Применение предобученных моделей, таких как CLIP и Talk2DINO, позволяет достичь высокой точности сопоставления текстовых и визуальных признаков, что критически важно для корректной идентификации объектов в сложных сценах. Метод поддерживает различные типы входных данных, включая как текстовые описания, так и изображения, что обеспечивает универсальность и адаптивность системы к различным сценариям использования. Кроме того, единое векторное представление признаков используется как для реконструкции сцены, так и для семантической сегментации, что способствует согласованности результатов и упрощает интеграцию всех компонентов в единую архитектуру.

Таким образом, модуль семантической сегментации с открытым словарём обеспечивает гибкий и эффективный способ выделения объектов в реконструированной сцене на основе произвольных текстовых запросов, что значительно расширяет возможности системы в целом.

3.7 Рендеринг объектов по запросу с новых ракурсов

Возможность рендеринга объектов с разных ракурсов по текстовому запросу открывает широкие перспективы для практических применений системы. Данная функциональность позволяет автоматически генерировать изображения найденных объектов под различными углами зрения, что имеет значительную ценность для решения множества задач компьютерного зрения и машинного обучения.

Одним из наиболее важных применений является расширение обучающих датасетов за счет автоматической генерации дополнительных изображений объектов, что позволяет увеличить разнообразие данных и улучшить обобщающую способность моделей машинного обучения. Кроме того, система обеспечивает формирование контролируемых синтетических наборов данных с известной геометрией и семантикой для тестирования и валидации алгоритмов.

Алгоритм 3.3 начинается с загрузки реконструированной сцены и вычисления семантического сходства между всеми гауссианами и текстовым запросом. Для этого используются языковые признаки гауссиан и эмбеддинг текстового запроса, сжатый с помощью PCA. Значения сходства нормализуются в диапазон [0, 1], где 1 соответствует максимальному сходству.

Алгоритм 3.3 Рендеринг объектов по запросу с новых ракурсов

Вход: Текстовый запрос Q_{text}

Вход: Реконструированная 3D-сцена $\mathcal{G} = \{g_i\}_{i=1}^{N}$, где $g_i = (\mu_i, s_i, R_i, c_i, \alpha_i, f_i)$

Вход: Модели CLIP, Talk2DINO, PCA

Вход: Пороги: $\tau_{sim} = 0.95$, параметры DBSCAN: $\epsilon, min_samples$

Вход: Параметры траектории: радиус орбиты r_{orbit}, количество ракурсов N_{views}

Выход: Набор изображений объекта $\mathcal{I}_{views} = \{I_j\}_{j=1}^{M}$

$E_{clip} \leftarrow \text{CLIP}(Q_{text})$ — Обработка текстового запроса: 1×512

$E_{dino} \leftarrow \text{Talk2DINO}(E_{clip})$ — Выравнивание пространства признаков: 1×768

$E_{query} \leftarrow \text{PCA}(E_{dino})$ — Сжатие эмбеддинга: $1 \times K$

$\mathcal{L} \leftarrow \{l_i\}_{i=1}^{N}$ — Извлечение семантических признаков гауссиан

$\mathcal{P} \leftarrow \{\mu_i\}_{i=1}^{N}$ — Извлечение 3D-позиций гауссиан

для каждая гауссиана $g_i \in \mathcal{G}$

  $s_i \leftarrow \text{cosine_similarity}(l_i, E_{query})$ — Косинусное сходство

$\mathcal{G}_{filtered} \leftarrow \{g_i : s_i > \tau_{sim}\}$ — Фильтрация по порогу сходства

$\mathcal{P}_{filtered} \leftarrow \{\mu_i : g_i \in \mathcal{G}_{filtered}\}$

если $|\mathcal{G}_{filtered}| > 0$ **то**

  $\mathcal{C} \leftarrow \text{DBSCAN}(\mathcal{P}_{filtered}, \epsilon, min_samples)$

  $\mathcal{I}_{views} \leftarrow \emptyset$

  для каждый кластер $C_k \in \mathcal{C}$

    $\mu_{center} \leftarrow \text{mean}(C_k)$

    Семплировать N_{views} точек на сфере радиуса r_{orbit} вокруг μ_{center}

    Рендерить сцену с каждой точки и сохранить видимые изображения

return $\mathcal{I}_{views}$

Гауссианы с высоким семантическим сходством (превышающим порог 0.95) группируются в кластеры с помощью алгоритма DBSCAN. Параметры кластеризации подобраны для эффективного выделения отдельных объектов в сцене. Для каждого найденного кластера вычисляется его центр как среднее положение входящих в него гауссиан.

Вокруг каждого центра генерируется сферическая траектория камеры с фиксированным радиусом. Траектория создается в плоскости XY с фиксированным углом наклона, что обеспечивает обзор объекта с уровня глаз. Для каждой позиции камеры на траектории выполняется рендеринг сцены, при этом проверяется глубина в центре

кадра. Кадры, где объект не перекрывается другими элементами сцены (глубина до объекта приблизительно равна радиусу орбиты), сохраняются в набор изображений.

4 Экспериментальная часть

В данном разделе представлены результаты оценки эффективности разработанного метода LEG-SLAM в задаче реконструкции 3D-сцен с открытой семантикой. Эксперименты включают оценку качества 3D-реконструкции на датасете Replica, анализ точности семантической сегментации на ScanNet с использованием открытой семантики, исследование SLAM-показателей (точность траектории камеры, скорость трекинга) и анализ влияния различных компонентов системы. Особое внимание уделяется сравнению с существующими методами по качеству реконструкции (PSNR, SSIM), точности семантической сегментации (mIoU, mAcc) и вычислительной эффективности (FPS, потребление памяти). Результаты демонстрируют преимущества предложенного подхода в контексте одновременной реконструкции и семантической сегментации 3D-сцен в реальном времени.

4.1 Датасеты

Для комплексной оценки предложенного метода LEG-SLAM в задачах 3D-реконструкции и семантической сегментации с открытым словарем используются два эталонных датасета: Replica и ScanNet. Выбор этих датасетов обусловлен наличием точных поз камеры для оценки SLAM-компонента, разнообразием сцен для проверки обобщающей способности, различными условиями съемки для анализа устойчивости к шуму, а также доступностью семантических аннотаций для количественной оценки сегментации.

ScanNet представляет реальные условия с шумом сенсоров и сложной геометрией, в то время как Replica обеспечивает контролируемую среду с идеальными данными для анализа верхних границ производительности.

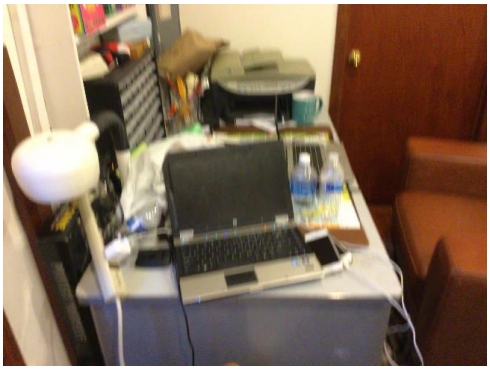
ScanNet [42] представляет собой большой RGB-D датасет реальных внутренних помещений, содержащий более 1500 сцен с высокоточными 3D-позами камер и детальными семантическими аннотациями. Каждая сцена представляет собой последовательность RGB-D кадров с разрешением 640×480 пикселей, снятых при естественном освещении с частотой 30 FPS. Позы камеры получены с помощью системы BundleFusion [43], обеспечивающей высокую точность локализации.

Семантические аннотации включают 20 классов объектов интерьера, представленных в Таблице 2.

Особенностью ScanNet является наличие реалистичных шумов и артефактов,

Таблица 1: Сравнительные характеристики используемых датасетов

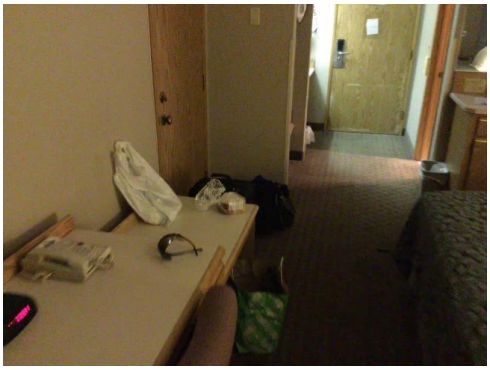
Характеристика	**ScanNet**	**Replica**
Количество сцен в эксперименте	12	8
Среднее количество кадров	1800	2000
Тип данных	Реальные RGB-D	Синтетические
Разрешение изображений	640×480	1200×680
Семантические аннотации	20 классов	Отсутствуют
Качество данных глубины	Шумные (Kinect)	Идеальные (GT)
Позы камеры	BundleFusion	Точные
Освещение	Естественное	Контролируемое
Сложность геометрии	Высокая	Средняя



(a) scene0050_02



(b) scene0144_01



(c) scene0221_01



(d) scene0300_01

Рис. 8: Примеры сцен из датасета ScanNet [42]

характерных для RGB-D сенсоров: пропуски в данных глубины на отражающих и прозрачных поверхностях, временная нестабильность измерений, окклюзии и изменения освещения. Это делает датасет идеальным для оценки устойчивости алгоритмов к реальным условиям съемки.

В настоящих экспериментах используется валидационное подмножество из

Таблица 2: Семантические классы датасета ScanNet

Структурные элементы	**Мебель**	**Предметы**	**Сантехника**
wall (стена)	cabinet (шкаф)	curtain (занавес)	refrigerator (холодильник)
floor (пол)	bed (кровать)	television (телевизор)	toilet (туалет)
ceiling (потолок)	chair (стул)	lamp (лампа)	sink (раковина)
door (дверь)	sofa (диван)	bag (сумка)	
window (окно)	table (стол)	person (человек)	
	shelves (полки)		
	counter (прилавок)		

12 сцен: scene0050_02, scene0144_01, scene0221_01, scene0300_01, scene0354_00, scene0389_00, scene0423_02, scene0427_00, scene0494_00, scene0616_00, scene0645_02, scene0693_00. Каждая сцена содержит в среднем 1800 кадров. Такой выбор обеспечивает разнообразие по геометрической сложности, освещению и плотности объектов.

Replica [44] — высококачественный синтетический датасет, содержащий фотореалистичные 3D-реконструкции реальных помещений, созданные с использованием современных техник трассировки лучей и физически корректного рендеринга.

Датасет включает 8 различных сцен высокого разрешения (1200×680 пикселей), воспроизводящих реальные помещения с высокой детализацией геометрии, материалов, освещения и текстур. Сцены охватывают два типа интерьеров: офисные помещения и комнаты.

Ключевым преимуществом Replica является отсутствие шума в данных: карты глубины являются точными (ground truth), позы камеры известны с машинной точностью, а освещение контролируется и воспроизводимо. Это позволяет проводить точную оценку алгоритмов реконструкции без влияния артефактов сенсоров и получать верхние оценки производительности.

Каждая сцена содержит в среднем 2000 кадров, снятых по заранее определенным траекториям, обеспечивающим полное покрытие пространства. Траектории камеры спроектированы для имитации естественного движения человека с учетом эргономических ограничений.

ScanNet используется для оценки производительности в реальных условиях с шумными данными RGB-D сенсоров, изменяющимся освещением и окклюзиями. Семантические аннотации на 20 классах объектов позволяют количественно измерить



(a) office_0



(b) office_2



(c) room_0



(d) room_2

Рис. 9: Примеры сцен из датасета Replica [44]

точность сегментации.

Replica применяется для анализа качества 3D-реконструкции в идеальных условиях с точными данными глубины и известными позами камеры. Датасет позволяет оценить верхние границы производительности системы и провести качественный анализ открытой семантической сегментации на произвольных текстовых запросах.

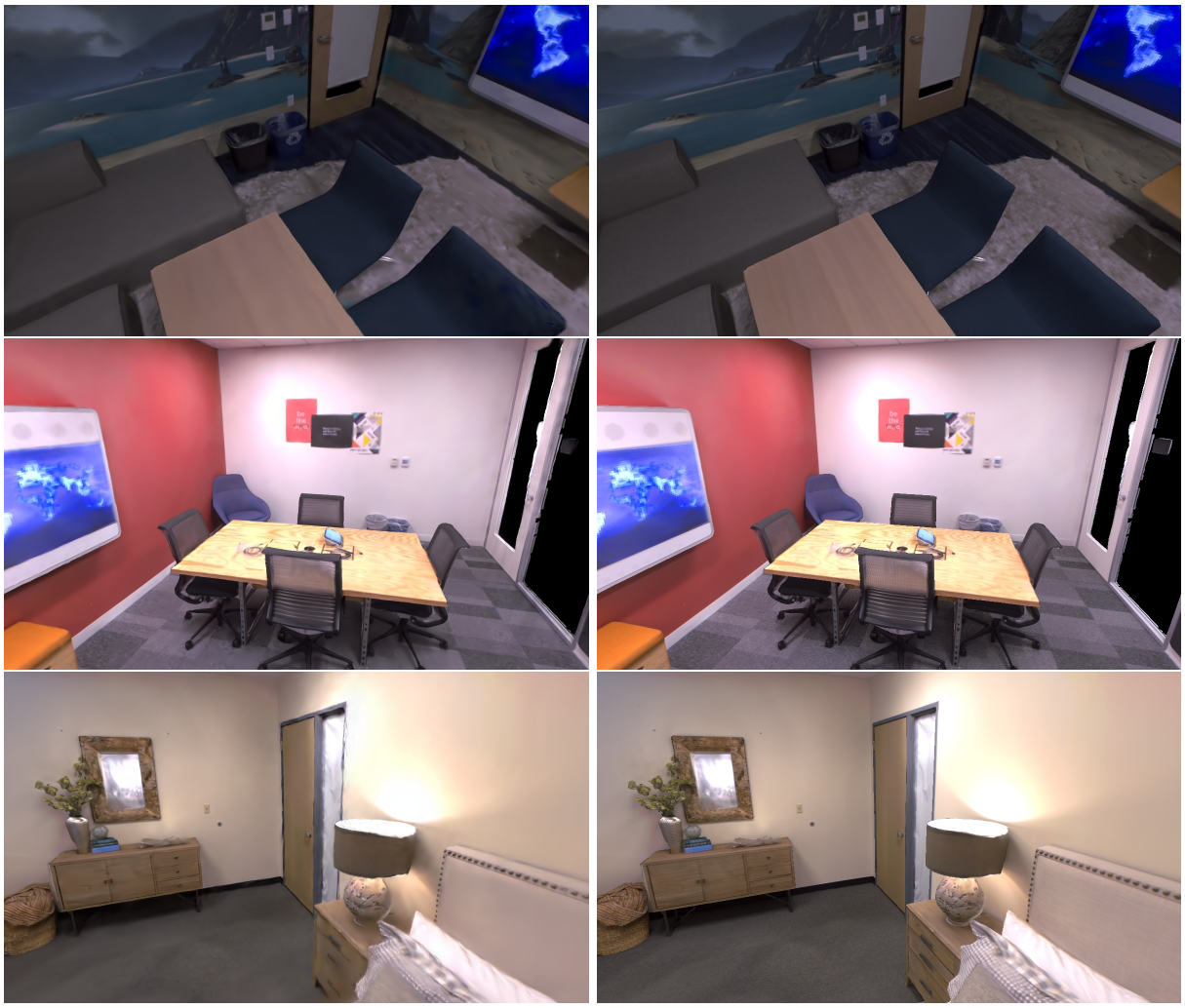
4.2 Визуализация результатов экспериментов

Визуализация качества реконструкции

На Рис. 10 приведено сравнение между реконструированным изображением сцены и исходным входным изображением. Видно, что результат реконструкции обладает фотореалистичным качеством и практически неотличим от оригинального изображения, что подтверждает высокую точность восстановления геометрии и текстур предлагаемого метода.

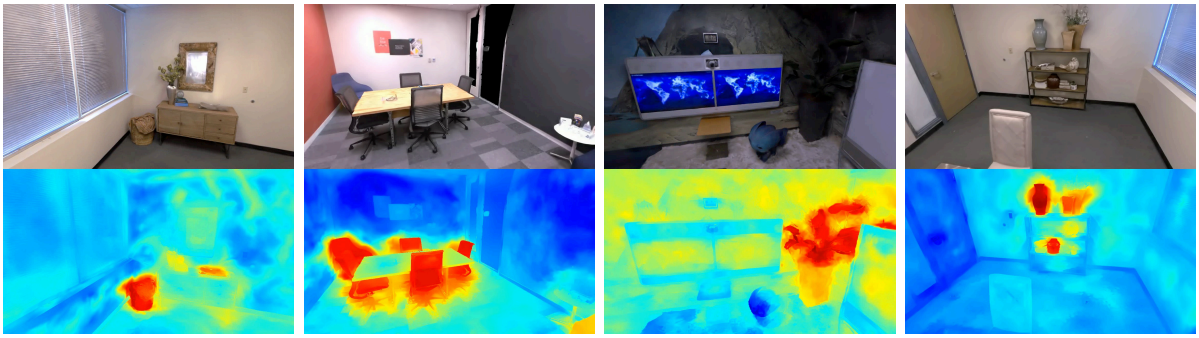
Визуализация семантической сегментации

На Рис. 11 приведены примеры работы LEG-SLAM по сегментации объектов по текстовым запросам в условиях открытой семантики. В каждом столбце сверху показан RGB-рендер сцены, снизу — соответствующая семантическая тепловая карта,



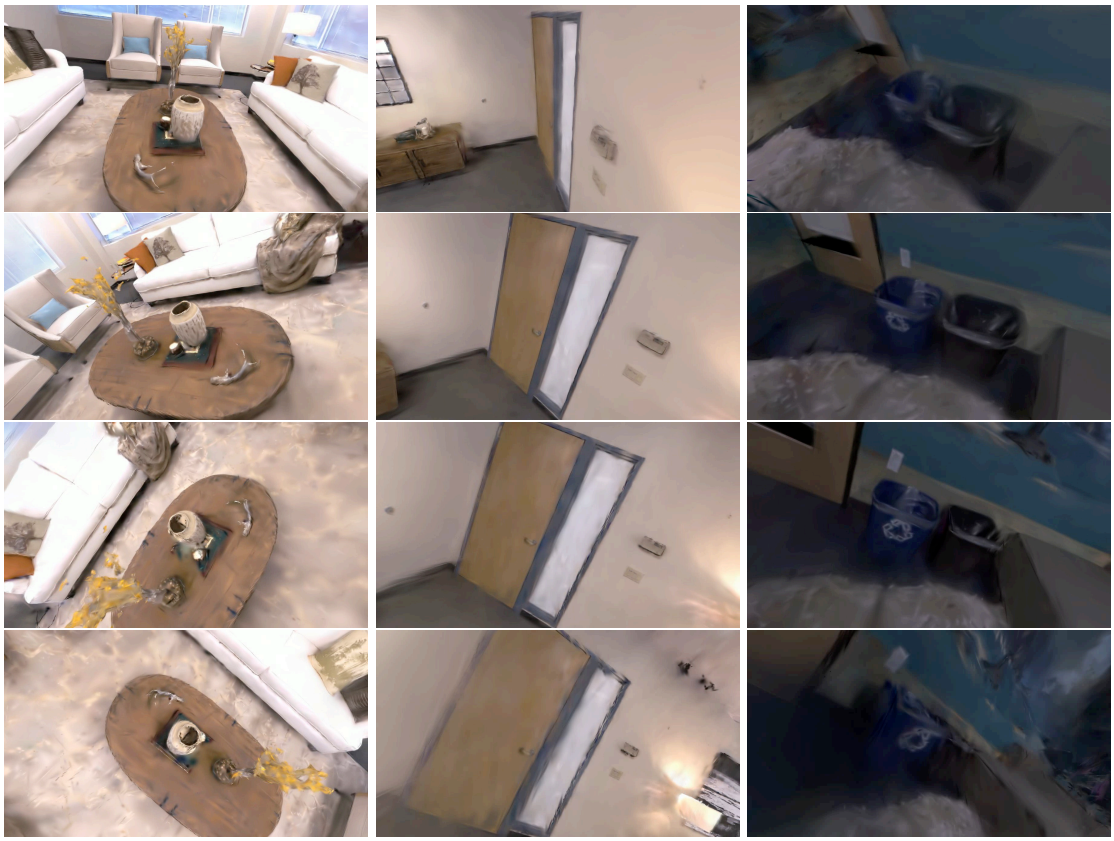
a) Реконструкция **b) Входное изображение**

Рис. 10: Сравнение реконструкции и входного изображения.



(a) Text query: "basket" (b) Text query: "chair" (c) Text query: "plant" (d) Text query: "vase"

Рис. 11: Визуализация RGB-изображения и карты семантической сегментации для различных текстовых запросов.



a) Ваза **b) Дверь** **c) Ведро**

Рис. 12: Визуализация рендеринга объектов с новых ракурсов.

отражающая вероятность принадлежности пикселей к запрошенному объекту, где красный цвет — высокая вероятность, синий — низкая.

(a) basket: Модель точно выделяет корзину, несмотря на наличие других предметов.

(b) chair: При запросе “chair” система уверенно локализует стулья, даже если они частично перекрыты столом. Семантическая карта хорошо ограничивает область стульев, игнорируя фон и соседние объекты.

(c) plant: Для запроса “plant” тепловая карта демонстрирует высокую активацию в области растения, несмотря на сливающийся фон и наличие других элементов.

(d) vase: В случае с “vase” LEG-SLAM корректно выделяет вазы. Тепловая карта показывает чёткую локализацию даже вблизи других объектов.

Модуль успешно справляется с неоднозначными сценами, выделяя только релевантные области, что подтверждает её пригодность для задач открытой семантики в 3D-реконструкции.

Рендеринг объектов с новых ракурсов

На Рис. 12 представлены примеры рендеринга объектов с ранее не встречавшихся ракурсов. Каждый столбец соответствует отдельному объекту (a — ваза, b — дверь, c — ведро), а строки показывают изображения этого объекта, полученные с разных точек обзора. Хорошо видно, что качество рендеринга зависит от положения камеры: если ракурс близок к тем, что использовались при обучении, изображение получается более реалистичным и детализированным. При удалении от обучающих траекторий появляются артефакты и снижается качество визуализации.

4.3 Сравнение с существующими методами

Детали реализации

Все эксперименты проводились на GPU NVIDIA RTX 4090 с 24 ГБ видеопамяти. В качестве основного экстрактора визуальных признаков использовалась модель DINOv2 ViT-B/14, а для обработки текстовых запросов — CLIP ViT-B/16. Проекция текстовых эмбеддингов в визуальное пространство осуществлялась с помощью модуля Talk2DINO

Сравнение качества 3D-реконструкции

Для оценки качества 3D-реконструкции были проведены эксперименты на датасете Replica. В Таблице 3 приведены результаты сравнения нашего метода с существующими SLAM-подходами.

LEG-SLAM демонстрирует конкурентоспособное качество реконструкции, немного уступая SOTA методам в качестве рендеринга. Предложенный метод превосходит остальные решения по скорости и обеспечивает открытую семантическую сегментацию, что делает его единственным методом, позоляющим использовать SLAM и открытую семантику в реальном времени. Также LEG-SLAM превосходит базовый метод Photo-SLAM по основным метрикам, кроме скорости обработки, так как добавление семантики приводит к замедлению общей скорости работы системы.

Сравнение методов сегментации по запросам с открытым словарем

В данном разделе проводится сравнение LEG-SLAM с современными методами сегментации по запросам с открытым словарем на датасете ScanNet. Оценка выпол-

Таблица 3: Количественное сравнение точности SLAM, скорости, качества рендеринга и наличия семантики. Результаты усреднены по 8 сценам из датасета Replica.

Method	ATE RMSE ↓	FPS ↑	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Semantic Vocabulary
NICE-SLAM [19]	2.51	0.198	24.42	0.809	0.233	-
Vox-Fusion [20]	1.47	1.07	24.41	0.801	0.236	-
Co-SLAM [21]	1.06	6.41	30.24	0.939	0.252	-
ESLAM [22]	0.62	3.02	29.08	0.929	0.239	-
SplaTAM [23]	**0.41**	0.21	**34.11**	**0.968**	**0.102**	-
Gaussian-SLAM [26]	**0.31**	0.63	42.08	**0.996**	**0.018**	-
MonoGS [27]	0.79	0.445	37.50	0.96	0.07	-
NEDS-SLAM [36]	0.354	-	34.76	0.962	0.088	Closed
RGBDS-SLAM [30]	0.589	-	38.85	0.967	0.035	Closed
SNI-SLAM [35]	0.456	2.15	-	-	-	Closed
GS^{3}SLAM [28]	0.37	-	36.26	0.989	0.052	Closed
SGS-SLAM [32]	0.412	2.11	34.15	0.97	0.096	Closed
OVO-Gaussian-SLAM [29]	**0.31**	0.28	**42.08**	**0.996**	**0.018**	**Open**
Photo-SLAM [37]	1.24	27.35	30.91	0.883	0.143	-
LEG-SLAM (Ours)	0.94	**10.56**	32.12	0.9151	0.1456	**Open**

Таблица 4: Сравнение методов open-vocabulary сегментации на 12 сценах ScanNet (в среднем 1800 кадров на сцену)

Method	Backbone	mIoU, %	mAcc, %	Training Time
LERF [7]	NeRF+CLIP	31.2	61.7	45 min
PVLFF [8]	NeRF+LSeg	52.9	67.0	65 min
LangSplat [10]	3DGS+CLIP	24.7	42.0	180 min
Feature3DGS [14]	3DGS+LSeg	59.2	75.1	150 min
Semantic Gaussians [11]	3DGS+LSeg	**62.0**	**77.0**	90 min
LEG-SLAM	3DGS+DINOv2	41.4	74.3	**1.5 min**

няется по двум основным метрикам: средняя пересекающаяся площадь (mIoU, %) и средняя точность по классам (mAcc, %), а также по времени обработки одной сцены.

Анализ результатов показывает, что методы можно разделить на две основные группы. Методы на основе NeRF (LERF, PVLFF) демонстрируют средние показатели точности (mIoU 31.2-52.9%), но требуют значительного времени на обучение (45-65 минут на сцену). В свою очередь, методы на основе 3D Gaussian Splatting показывают более высокую точность (mIoU 24.7-62.0%), однако время обучения варьируется от 90 до 180 минут на сцену.

LEG-SLAM достигает mIoU 41.4% и mAcc 74.3%, что немного уступает лучшим результатам по mIoU. Однако ключевым преимуществом является вычисли-

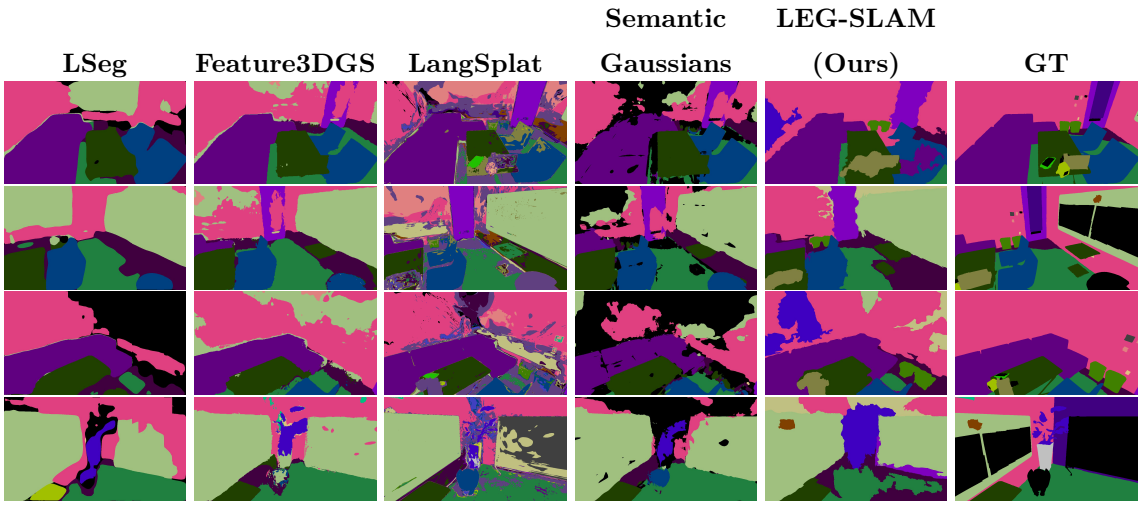


Рис. 13: Сравнение качества семантической сегментации различных методов на датасете Replica.

тельная эффективность: среднее время обработки одной сцены составляет всего 1.5 минуты (18 FPS), что в 30-120 раз быстрее других методов. Это делает LEG-SLAM единственным подходом, способным работать в реальном времени при сохранении конкурентоспособного качества сегментации.

На Рис. 13 представлены визуальные примеры сегментации для различных методов. Анализ показывает, что существующие подходы часто страдают от шумовых предсказаний и артефактов, особенно в сложных сценах.

Сравнение SLAM-систем

Оценка производительности LEG-SLAM в задаче SLAM проводится на датасете Replica. Основные метрики сравнения включают точность траектории (ATE RMSE), качество рендеринга (PSNR, SSIM) и скорость работы (FPS).

Анализ результатов показывает, что существующие SLAM-системы можно разделить на две категории. Точные, но медленные методы достигают высокого качества реконструкции, но работают значительно медленнее реального времени. В свою очередь, быстрые, но менее точные методы обеспечивают работу в реальном времени, но уступают по качеству реконструкции.

LEG-SLAM демонстрирует сбалансированные показатели: конкурентоспособное качество реконструкции (PSNR 32.12, SSIM 0.9151), высокую скорость работы (10.56 FPS) и возможность открытой семантической сегментации в реальном времени.

По сравнению с базовым методом Photo-SLAM, LEG-SLAM показывает улучшение по всем метрикам качества. Единственное снижение наблюдается в скорости

Таблица 5: Сравнение моделей по метрикам качества генерации текстового описания изображения.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	METEOR	HAR	TIME
Ferret	0.13	0.08	0.06	0.03	0.26	0.31	0.06	2.61s
LLaVA	**0.68**	**0.57**	**0.51**	0.40	**0.77**	**0.67**	**0.94**	0.72s
Kosmos	0.23	0.18	0.18	0.14	0.35	0.28	0.33	3.0s
InternLM-2 VL 1.8B	0.56	0.44	0.42	0.36	0.68	0.51	0.69	**0.41s**
InternLM-2 7B	0.61	0.51	0.50	**0.46**	0.74	0.62	0.81	0.63s
InternLM-2 7B-4bit	0.59	0.49	0.47	0.44	0.71	0.60	0.75	0.58s
InternLM-2 VL 7B	0.37	0.19	0.17	0.12	0.62	0.28	0.72	0.93s
InternLM-2 VL 7B-4bit	0.44	0.29	0.27	0.24	0.51	0.36	0.64	0.71s

работы, что объясняется добавлением новых модулей для семантической сегментации. Однако, даже с учетом этого снижения, система сохраняет способность работать в реальном времени.

Небольшое снижение метрик качества по сравнению с лучшими методами компенсируется значительным приростом в скорости и добавлением функциональности семантической сегментации. Это делает LEG-SLAM единственным методом, сочетающим real-time SLAM с открытой семантикой, что особенно важно для практических применений.

4.4 Анализ влияния компонентов метода

Сравнение методов генерации описаний

Для автоматического создания текстовых описаний изображений была выбрана модель LLaVa. Этот выбор основан на сравнительном анализе различных современных моделей (таблица 5).

Полученные в ходе наших экспериментов результаты по качеству генерации описаний с помощью LLaVA согласуются с выводами работы [45], где эта модель также была выбрана как наиболее эффективная для captioning объектов в indoor-сценах.

LLaVa показывает наилучшие результаты по качеству генерируемых описаний. Особенно важным является показатель HAR (Human Agreement Rate) равный 0.94, который означает, что в 94% случаев описания, созданные моделью, совпадают с тем, как объекты воспринимают люди. Это критически важно для точной сегментации объектов в сцене.

Кроме высокого качества описаний, LLaVa обеспечивает достаточно быструю работу — время обработки одного изображения составляет 0.72 секунды. Это позволяет

использовать модель в системах реального времени без существенного замедления общей производительности.

Альтернативные модели либо показывают значительно худшее качество описаний, либо работают слишком медленно, что делает их непригодными для данной задачи.

Сравнение архитектур извлечения признаков

Выбор архитектуры для извлечения визуальных признаков существенно влияет на качество семантической реконструкции и возможность работы в реальном времени. В рамках экспериментов были протестированы два подхода: DINOv2 ViT-B/14 и OpenSeg.

DINOv2 представляет собой самообучающуюся модель, формирующую универсальные эмбеддинги изображений без привязки к фиксированному набору категорий. Это позволяет ей хорошо обобщаться на разнообразные сцены и объекты, включая ранее не встречавшиеся классы. Такие свойства особенно важны при построении open-vocabulary семантических карт.

OpenSeg, в свою очередь, представляет собой специализированную модель, ориентированную на задачу семантической сегментации и обученную преимущественно на наборах категорий, таких как COCO. Несмотря на высокую точность на знакомых классах, OpenSeg демонстрирует ограниченную способность к обобщению вне обучающего словаря.

Таким образом, DINOv2 обеспечивает более устойчивое извлечение признаков в условиях открытых словарей и является предпочтительным выбором для семантически обогащённой реконструкции в реальном времени.

Для оценки вычислительной эффективности было измерено время обработки, включая сжатие эмбеддингов с помощью PCA. OpenSeg + PCA требует 100 мс, а DINOv2 + PCA — всего 33 мс (ускорение в 3 раза). Это делает DINOv2 предпочтительным выбором для задач реального времени.

Влияние методов сжатия семантических признаков

Результаты на рисунке 14 демонстрируют, что увеличение размерности семантических признаков положительно влияет на косинусное сходство, однако начиная с порога 64 признаков прирост становится несущественным. При этом дальнейшее

Таблица 6: Сравнение методов сжатия эмбеддингов (все методы сжимают до размерности64)

Method	**CosSim ↑**	**mIoU ↑**	**Time (ms) ↓**
MLP-AE (2-layer)	0.9379	37.3	0.85
MLP-AE (4-layer)	0.9551	34.8	2.96
PCA	**0.9570**	**41.2**	**0.60**

Таблица 7: Сравнение размерности сжатия эмбеддингов с помощью PCA

Dim	**Cos. Sim.↑**	**mIoU↑**	**PSNR↑**	**Train FPS↑**
3	0.7423	6.6	24.34	18.38
6	0.8348	20.1	**25.07**	**18.81**
9	0.8814	27.2	24.83	18.72
12	0.9070	32.0	24.53	18.63
16	0.9298	37.1	24.7	18.4
24	0.9532	39.1	24.75	18.22
32	0.9662	40.1	24.48	18.09
48	0.9808	40.4	24.30	17.52
64	0.9880	41.4	24.26	17.55
80	0.9920	42.2	23.96	17.63
96	0.9946	42.3	23.65	17.58
128	**0.9973**	**42.7**	23.57	17.44

увеличение размерности приводит к росту времени инференса, нагрузке на видеопамять и усложнению оптимизации параметров гауссианов, что негативно сказывается на качестве реконструкции.

При использовании чрезмерно малых размерностей (например, 3) наблюдается агрессивное сжатие, сопровождающееся потерей критически важной семантической информации. Это затрудняет оптимизацию сцены и приводит к неэффективному использованию гауссиан, увеличивая их количество без реального прироста качества.

На основании проведённого анализа была выбрана размерность семантических признаков равная 64, как обеспечивающая оптимальный компромисс между точностью семантической реконструкции, эффективностью оптимизации сцены и вычислительной нагрузкой.

Влияние размерности эмбеддингов

Одним из ключевых гиперпараметров системы LEG-SLAM является размерность эмбеддингов после сжатия методом главных компонент (PCA). Размерность напрямую влияет как на полноту сохранённой семантической информации, так и на вычислительную эффективность, объём видеопамяти и сложность последующей оптимизации параметров сцены.

Для анализа влияния размерности были проведены два типа экспериментов: (1) оценка качества восстановления признаков по метрике косинусного сходства между исходными и сжатыми эмбеддингами, и (2) комплексная оценка всей системы по метрикам mIoU (семантика), PSNR (геометрия/визуальное качество) и FPS (производительность).

В первом эксперименте случайным образом выбиралось по 10 кадров из каждой из 12 сцен набора ScanNet. Для каждого кадра извлекались признаки DINOv2 (размерность 648×484×768), после чего они сэмплировались с шагом 4 для повышения разнообразия. Для каждого значения целевой размерности (от 3 до 128) обучался собственный PCA-проектор, и вычислялось среднее косинусное сходство между оригинальными и восстановленными эмбеддингами.

Во втором эксперименте система LEG-SLAM запускалась на полных сценах ScanNet с различными размерностями сжатия. Измерялись значения mIoU, PSNR и FPS, а также потребление видеопамяти.

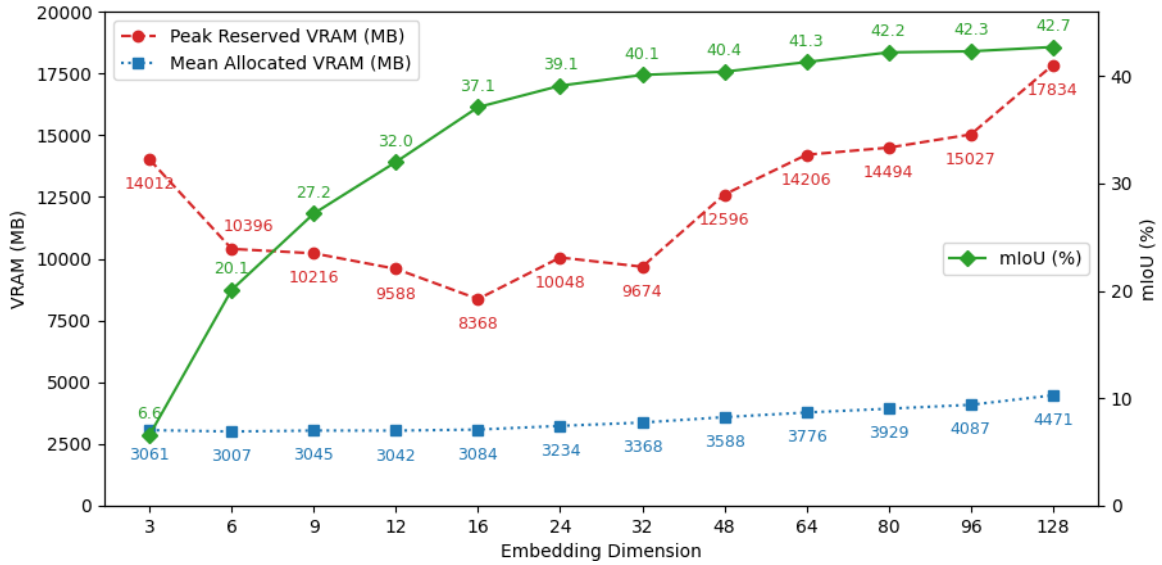
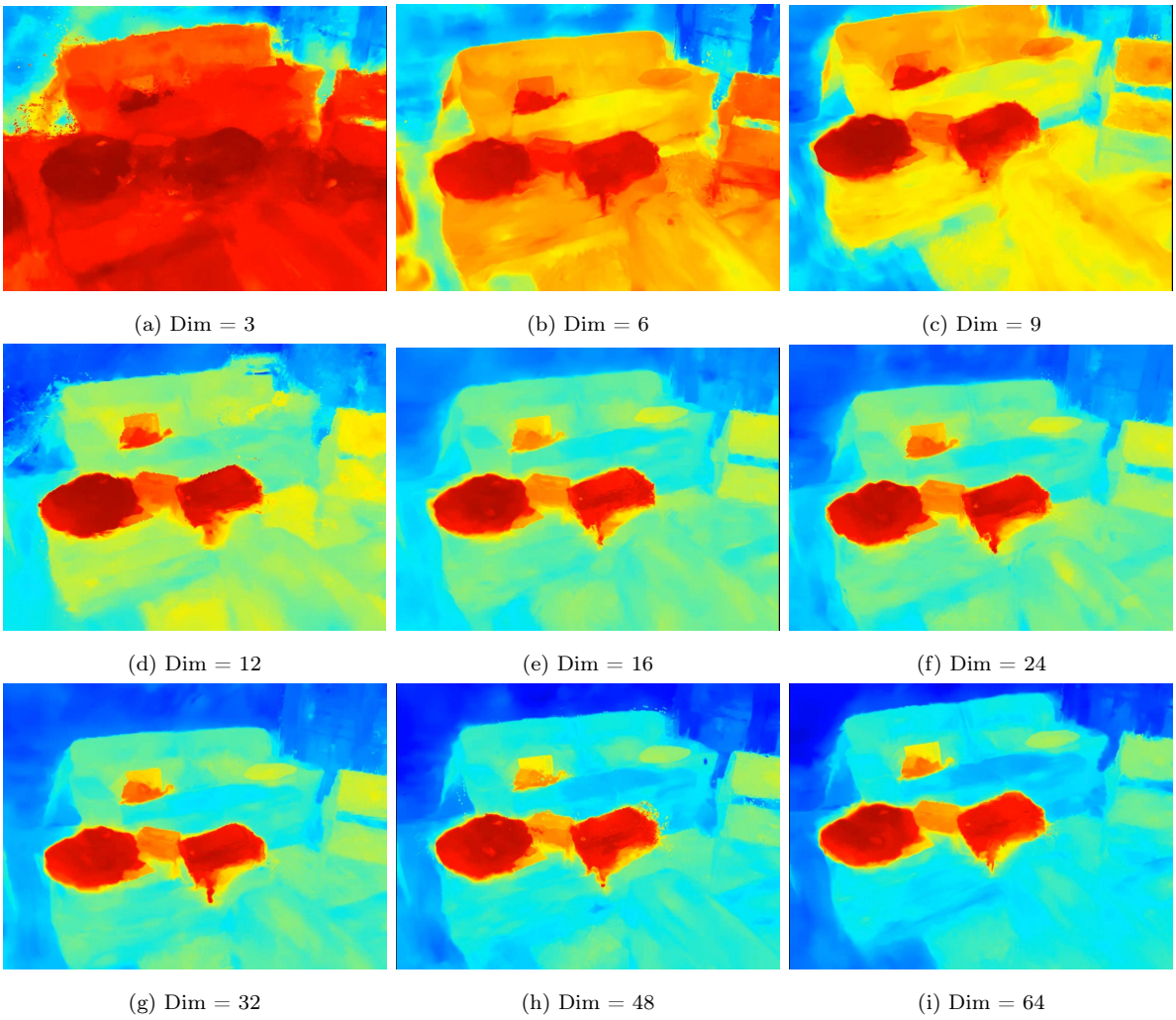


Рис. 14: Потребление видеопамяти при разных размерностях эмбеддингов



(a) Dim = 3 (b) Dim = 6 (c) Dim = 9

(d) Dim = 12 (e) Dim = 16 (f) Dim = 24

(g) Dim = 32 (h) Dim = 48 (i) Dim = 64

Рис. 15: Сравнение качества сегментации для запроса **backpack** при разных размерностях семантических признаков.

На рисунке 15 представлены примеры сегментации для запроса **backpack** при разных размерностях семантических признаков. Видно, что при увеличении размерности признаков качество сегментации постепенно улучшается, при это на больших размерностях прирост замедляется.

Рисунок 14 показывает зависимость потребления видеопамяти и метрики mIoU от размера вектора семантических признаков. Стоит отметить, что при низких размерностях наблюдается агрессивное сжатие, сопровождающееся потерей критически важной семантической информации. Это приводит к неэффективному использованию гауссианов, увеличивая их количество и ухудшая качество сегментации. При слишком больших размерностях наблюдается рост потребления видеопамяти, при этом прирост качества сегментации становится несущественным.

На основании проведённого анализа была выбрана размерность эмбеддингов

равная 64, как обеспечивающая оптимальный компромисс между точностью семантической реконструкции, эффективностью оптимизации сцены и вычислительной нагрузкой.

5 Заключение

В результате выполнения научно-исследовательской работы были получены следующие результаты:

- Предложен способ интеграции семантических признаков DINOv2 в метод гауссовского сплаттинга
- Разработан эффективный модуль сжатия семантических признаков на основе PCA
- Проведен анализ оптимальной размерности сжатия семантических признаков, в результате которого выбрана размерность, обеспечивающая оптимальный баланс между качеством и производительностью
- Реализован подход, поддерживающий произвольные текстовые запросы в сцене и работающий в режиме реального времени

Эксперименты на датасетах Replica и ScanNet продемонстрировали значительные преимущества предложенного метода LEG-SLAM. Система обеспечивает конкурентоспособное качество реконструкции при работе в реальном времени. По сравнению с базовым методом Photo-SLAM наблюдается улучшение всех метрик качества, что подтверждает эффективность предложенных решений. Ключевым преимуществом является уникальная возможность открытой семантической сегментации в реальном времени, недоступная в других существующих методах. При этом LEG-SLAM демонстрирует значительное превосходство в скорости работы, превосходя существующие методы открытой семантической сегментации в 30-120 раз.

Дальнейшие исследования предполагается посвятить повышению качества визуальной реконструкции и улучшению точности семантической сегментации в сложных сценах с динамическими объектами.

Список литературы

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi и Ren Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. arXiv:2003.08934 [cs]. Авг. 2020. DOI: 10.48550/arXiv.2003.08934. URL: http://arxiv.org/abs/2003.08934 (дата обр. 22.12.2024).

[2] Thomas Müller, Alex Evans, Christoph Schied и Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding". en. В: *ACM Transactions on Graphics* 41.4 (июль 2022), с. 1—15. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3528223.3530127. URL: https://dl.acm.org/doi/10.1145/3528223.3530127 (дата обр. 13.06.2025).

[3] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister и Angjoo Kanazawa. "Nerfstudio: A Modular Framework for Neural Radiance Field Development". В: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*. arXiv:2302.04264 [cs]. Июль 2023, с. 1—12. DOI: 10.1145/3588432.3591516. URL: http://arxiv.org/abs/2302.04264 (дата обр. 13.06.2025).

[4] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz и Ricardo Martin-Brualla. *Nerfies: Deformable Neural Radiance Fields*. arXiv:2011.12948 [cs]. Сент. 2021. DOI: 10.48550/arXiv.2011.12948. URL: http://arxiv.org/abs/2011.12948 (дата обр. 13.06.2025).

[5] Ajay Jain, Matthew Tancik и Pieter Abbeel. "Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis". en. В: 2021, с. 5885—5894. URL: https://openaccess.thecvf.com/content/ICCV2021/html/Jain_Putting_NeRF_on_a_Diet_Semantically_Consistent_Few-Shot_View_Synthesis_ICCV_2021_paper.html (дата обр. 13.06.2025).

[6] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger и Andrew J. Davison. *In-Place Scene Labelling and Understanding with Implicit Scene Representation*. arXiv:2103.15875 [cs]. Авг. 2021. DOI: 10.48550/arXiv.2103.15875. URL: http://arxiv.org/abs/2103.15875 (дата обр. 22.12.2024).

[7] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa и Matthew Tancik. *LERF: Language Embedded Radiance Fields*. arXiv:2303.09553 [cs]. Март 2023. DOI: 10.48550/arXiv.2303.09553. URL: http://arxiv.org/abs/2303.09553 (дата обр. 23.12.2024).

[8] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai и Peter Kontschieder. *Panoptic Lifting for 3D Scene Understanding with Neural Fields*. arXiv:2212.09802 [cs]. Дек. 2022. DOI: 10.48550/arXiv.2212.09802. URL: http://arxiv.org/abs/2212.09802 (дата обр. 22.12.2024).

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler и George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering". en. В: *ACM Transactions on Graphics* 42.4 (авг. 2023), с. 1—14. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3592433. URL: https://dl.acm.org/doi/10.1145/3592433 (дата обр. 22.12.2024).

[10] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang и Hanspeter Pfister. "Langsplat: 3d language gaussian splatting". В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, с. 20051—20060. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Qin_LangSplat_3D_Language_Gaussian_Splatting_CVPR_2024_paper.html (дата обр. 22.12.2024).

[11] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu и Qing Li. *Semantic Gaussians: Open-Vocabulary Scene Understanding with 3D Gaussian Splatting*. arXiv:2403.15624 [cs]. Авг. 2024. DOI: 10.48550/arXiv.2403.15624. URL: http://arxiv.org/abs/2403.15624 (дата обр. 22.12.2024).

[12] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang и Yongdong Zhang. *MotionGS: Exploring Explicit Motion Guidance for Deformable 3D Gaussian Splatting*. arXiv:2410.07707 [cs]. Окт. 2024. DOI: 10.48550/arXiv.2410.07707. URL: http://arxiv.org/abs/2410.07707 (дата обр. 13.06.2025).

[13] Mihnea-Bogdan Jurca, Remco Royen, Ion Giosan и Adrian Munteanu. *RT-GS2: Real-Time Generalizable Semantic Segmentation for 3D Gaussian Representations*

of Radiance Fields. arXiv:2405.18033 [cs]. Авг. 2024. DOI: 10.48550/arXiv.2405.18033. URL: http://arxiv.org/abs/2405.18033 (дата обр. 22.12.2024).

[14] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang и Achuta Kadambi. "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields". В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, с. 21676—21685.

[15] Golnaz Ghiasi, Xiuye Gu, Yin Cui и Tsung-Yi Lin. "Scaling Open-Vocabulary Image Segmentation with Image-Level Labels". В: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*. Berlin, Heidelberg: Springer-Verlag, окт. 2022, с. 540—557. ISBN: 978-3-031-20058-8. DOI: 10.1007/978-3-031-20059-5_31. URL: https://doi.org/10.1007/978-3-031-20059-5_31 (дата обр. 07.03.2025).

[16] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun и René Ranftl. *Language-driven Semantic Segmentation*. arXiv:2201.03546 [cs]. Апр. 2022. DOI: 10.48550/arXiv.2201.03546. URL: http://arxiv.org/abs/2201.03546 (дата обр. 22.12.2024).

[17] Chong Zhou, Chen Change Loy и Bo Dai. *Extract Free Dense Labels from CLIP*. arXiv:2112.01071 [cs]. Июль 2022. DOI: 10.48550/arXiv.2112.01071. URL: http://arxiv.org/abs/2112.01071 (дата обр. 13.06.2025).

[18] Luca Barsellotti, Lorenzo Bianchi, Nicola Messina, Fabio Carrara, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi и Rita Cucchiara. *Talking to DINO: Bridging Self-Supervised Vision Backbones with Language for Open-Vocabulary Segmentation*. arXiv:2411.19331 [cs]. Нояб. 2024. DOI: 10.48550/arXiv.2411.19331. URL: http://arxiv.org/abs/2411.19331 (дата обр. 22.12.2024).

[19] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald и Marc Pollefeys. "Nice-slam: Neural implicit scalable encoding for slam". В: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, с. 12786—12796.

[20] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu и Guofeng Zhang. "Vox-Fusion: Dense Tracking and Mapping with Voxel-based Neural Implicit Representation". В: *2022 IEEE International Symposium on Mixed and Augmented*

Reality (ISMAR). arXiv:2210.15858 [cs]. Окт. 2022, с. 499—507. DOI: 10.1109/ISMAR55827.2022.00066. URL: http://arxiv.org/abs/2210.15858 (дата обр. 07.03.2025).

[21] Hengyi Wang, Jingwen Wang и Lourdes Agapito. *Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM*. arXiv:2304.14377 [cs]. Апр. 2023. DOI: 10.48550/arXiv.2304.14377. URL: http://arxiv.org/abs/2304.14377 (дата обр. 07.03.2025).

[22] Mohammad Mahdi Johari, Camilla Carta и Francois Fleuret. "ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields". en. В: *2023 Ieee/Cvf Conference On Computer Vision And Pattern Recognition (Cvpr)*. ISSN: 1063-6919. Los Alamitos, янв. 2023, с. 17408—17419. DOI: 10.1109/CVPR52729.2023.01670. URL: https://infoscience.epfl.ch/handle/20.500.14299/204022 (дата обр. 07.03.2025).

[23] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan и Jonathon Luiten. "Splatam: Splat track & map 3d gaussians for dense rgb-d slam". В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, с. 21357—21366.

[24] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang и Xuelong Li. "Gs-slam: Dense visual slam with 3d gaussian splatting". В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, с. 19595—19604.

[25] Seongbo Ha, Jiung Yeon и Hyeonwoo Yu. *RGBD GS-ICP SLAM*. _eprint: 2403.12550. 2024. URL: https://arxiv.org/abs/2403.12550.

[26] Vladimir Yugay, Yue Li, Theo Gevers и Martin R. Oswald. *Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting*. arXiv:2312.10070 [cs]. Март 2024. DOI: 10.48550/arXiv.2312.10070. URL: http://arxiv.org/abs/2312.10070 (дата обр. 02.06.2025).

[27] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly и Andrew J. Davison. *Gaussian Splatting SLAM*. arXiv:2312.06741 [cs]. Апр. 2024. DOI: 10.48550/arXiv.2312.06741. URL: http://arxiv.org/abs/2312.06741 (дата обр. 02.06.2025).

[28] Linfei Li, Lin Zhang, Zhong Wang и Ying Shen. “GS^3 LAM: Gaussian Semantic Splatting SLAM”. en. В: *Proceedings of the 32nd ACM International Conference on Multimedia*. Melbourne VIC Australia: ACM, окт. 2024, с. 3019—3027. ISBN: 9798400706868. DOI: 10.1145/3664647.3680739. URL: https://dl.acm.org/doi/10.1145/3664647.3680739 (дата обр. 22.12.2024).

[29] Tomas Berriel Martins, Martin R. Oswald и Javier Civera. *OVO-SLAM: Open-Vocabulary Online Simultaneous Localization and Mapping*. arXiv:2411.15043 [cs] version: 1. Нояб. 2024. DOI: 10.48550/arXiv.2411.15043. URL: http://arxiv.org/abs/2411.15043 (дата обр. 02.06.2025).

[30] Zhenzhong Cao, Chenyang Zhao, Qianyi Zhang, Jinzheng Guang и Yinuo Song Jingtai Liu. *RGBDS-SLAM: A RGB-D Semantic Dense SLAM Based on 3D Multi Level Pyramid Gaussian Splatting*. arXiv:2412.01217 [cs]. Дек. 2024. DOI: 10.48550/arXiv.2412.01217. URL: http://arxiv.org/abs/2412.01217 (дата обр. 02.06.2025).

[31] Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang и Kun Zhou. “RTG-SLAM: Real-time 3D Reconstruction at Scale using Gaussian Splatting”. en. В: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*. Denver CO USA: ACM, июль 2024, с. 1—11. ISBN: 9798400705250. DOI: 10.1145/3641519.3657455. URL: https://dl.acm.org/doi/10.1145/3641519.3657455 (дата обр. 22.12.2024).

[32] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng и Hongyu Wang. “SGS-SLAM: Semantic Gaussian Splatting For Neural Dense SLAM”. В: т. 15089. arXiv:2402.03246 [cs]. 2025, с. 163—179. DOI: 10.1007/978-3-031-72751-1_10. URL: http://arxiv.org/abs/2402.03246 (дата обр. 22.12.2024).

[33] Kunyi Li, Michael Niemeyer, Nassir Navab и Federico Tombari. *DNS SLAM: Dense Neural Semantic-Informed SLAM*. _eprint: 2312.00204. 2023. URL: https://arxiv.org/abs/2312.00204.

[34] Siting Zhu, Renjie Qin, Guangming Wang, Jiuming Liu и Hesheng Wang. *SemGauss-SLAM: Dense Semantic Gaussian Splatting SLAM*. _eprint: 2403.07494. 2024. URL: https://arxiv.org/abs/2403.07494.

[35] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys и Hesheng Wang. *SNI-SLAM: Semantic Neural Implicit SLAM*. arXiv:2311.11016 [cs]. Март 2024. DOI: 10.48550/arXiv.2311.11016. URL: http://arxiv.org/abs/2311.11016 (дата обр. 07.03.2025).

[36] Yiming Ji, Yang Liu, Guanghu Xie, Boyu Ma и Zongwu Xie. "NEDS-SLAM: A Neural Explicit Dense Semantic SLAM Framework using 3D Gaussian Splatting". В: *IEEE Robotics and Automation Letters* 9.10 (окт. 2024). arXiv:2403.11679 [cs], с. 8778—8785. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2024.3451390. URL: http://arxiv.org/abs/2403.11679 (дата обр. 15.05.2025).

[37] Huajian Huang, Longwei Li, Hui Cheng и Sai-Kit Yeung. "Photo-SLAM: Real-time Simultaneous Localization and Photorealistic Mapping for Monocular Stereo and RGB-D Cameras". en. В: 2024, с. 21584—21593. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Huang_Photo-SLAM_Real-time_Simultaneous_Localization_and_Photorealistic_Mapping_for_Monocular_Stereo_CVPR_2024_paper.html (дата обр. 22.12.2024).

[38] Roman Titkov, Egor Zubkov, Dmitry Yudin, Jaafar Mahmoud, Malik Mohrat и Gennady Sidorov. *LEG-SLAM: Real-Time Language-Enhanced Gaussian Splatting for SLAM*. arXiv:2506.03073 [cs]. Июнь 2025. DOI: 10.48550/arXiv.2506.03073. URL: http://arxiv.org/abs/2506.03073 (дата обр. 14.06.2025).

[39] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel и Juan D. Tardós. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM". В: *IEEE Transactions on Robotics* 37.6 (дек. 2021). arXiv:2007.11898 [cs], с. 1874—1890. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2021.3075644. URL: http://arxiv.org/abs/2007.11898 (дата обр. 13.06.2025).

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby и др. "DINOv2: Learning Robust Visual Features without Supervision". В: *Transactions on Machine Learning Research Journal* (2024), с. 1—31.

[41] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li и Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". В: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919. Июнь 2009, с. 248—255. DOI:

10.1109/CVPR.2009.5206848. URL: https://ieeexplore.ieee.org/abstract/document/5206848 (дата обр. 14.06.2025).

[42] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser и Matthias Nießner. “Scannet: Richly-annotated 3d reconstructions of indoor scenes”. В: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, с. 5828—5839.

[43] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi и Christian Theobalt. *BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration*. arXiv:1604.01093 [cs]. Февр. 2017. DOI: 10.48550/arXiv.1604.01093. URL: http://arxiv.org/abs/1604.01093 (дата обр. 06.06.2025).

[44] Julian Straub и др. *The Replica Dataset: A Digital Replica of Indoor Spaces*. arXiv:1906.05797 [cs]. Июнь 2019. DOI: 10.48550/arXiv.1906.05797. URL: http://arxiv.org/abs/1906.05797 (дата обр. 22.12.2024).

[45] Sergey Linok, Tatiana Zemskova, Svetlana Ladanova, Roman Titkov, Dmitry Yudin, Maxim Monastyrny и Aleksei Valenkov. *Beyond Bare Queries: Open-Vocabulary Object Grounding with 3D Scene Graph*. arXiv:2406.07113 [cs]. Май 2025. DOI: 10.48550/arXiv.2406.07113. URL: http://arxiv.org/abs/2406.07113 (дата обр. 14.06.2025).